University of Economics, Prague

Faculty of Informatics and Statistics

Department of Information and Knowledge Engineering

Ontology Tool Benchmarks and Their Construction

Habilitation thesis

ondrej.zamazal@vse.cz

Prague, Králův Dvůr, 2018

Preface

Since its inception in 1990, the World-Wide Web (WWW) has had a tremendous impact on the world. It happened in at least three different phases. The original Web 1.0 is based on static web pages and their hyperlinks and was mostly in a read-only mode. Web 2.0 concerns user-generated content, with users not only consuming data but also contributing information via blogs, vlogs, etc. From the technical point of view, while the web applications were already created during Web 1.0, in Web 2.0 users started to build rich web applications. These web applications are often based on service-oriented architecture and often are user-oriented (e.g., Facebook). While Web 2.0 realized one of the classic tenets of WWW, the AAA slogan ("Anyone can say Anything about Any topic"), it still shares the drawback of Web 1.0: the content is not seamlessly understandable by machines. In 2001, Web 3.0 was established and labeled "the semantic web", where information can be equipped with well-defined meaning. Basically, this is finally the web which should be understandable by machines, enabling a construction of smart web applications. Although the semantic web, or, specifically, the web of data, has not yet been realized in its full potential, the results are already being used world-wide. In this respect, ontologies, representing formal conceptual models typically describing a certain domain of discourse, are deployed in world-wide applications, such as Google Search where less expressive ontology, i.e. vocabulary, schema.org is used. Furthermore, many companies recognized ontologies as a suitable knowledge representation means to formally specify shared conceptualization. There is a growing number of ontologies and, on the other side, there are increasingly more ontology tools. Similar to other artefacts of informatics, ontology tools need to be benchmarked in order to assure a satisfactory level of quality.

The central theme of this thesis is an ontology benchmark[1] which represents a material

[1] Throughout this thesis, the notions "ontology benchmark", "ontology tool benchmark" and "benchmarking corpus" are used interchangeably.

for benchmarking of ontology tools. Although benchmarking activities and corresponding benchmarks are common to the semantic web field, they have not been uniformly characterized and summarized. In this thesis, I provide a categorization of ontology tools, which enables each category to be described, using eight activities which can be performed on ontologies. Each activity can be described using five characteristics. The relationships between categories, activities and characteristics are depicted in Figure 1.2. These concepts are explained in Section 1.1.1. Further, I provide the typical requirements of ontology benchmarks for each ontology tool category in Section 1.2. Next, I provide a survey of existing ontology benchmarks, grouped according to my ontology tool categorization and describe them using activities and characteristics from the categorization. Further, according to this (expert-based) survey, I provide a simple recommender suggesting suitable ontology benchmarks, according to a user's needs. This recommendation is based on ontology benchmark descriptions, using activities and their characteristics, instead of the predefined suitability of ontology benchmarks for a specific ontology tool category. The ontology tool categorization serves firstly for the grouping of ontology benchmarks and their description. Ontology benchmarks can be constructed manually or automatically. The thesis provides an example of a manually created, widely-used ontology benchmark, OntoFarm, and the description of the tool supporting an automatic construction of ontology benchmarks. Both, OntoFarm and the tool, are considered as contributions of the author to the field.

Although the work described here has been carried out in several different projects, it has mainly been performed within the post-doctoral project supported by the Czech Science Foundation grant, no. 14-14076, "COSOL – Categorization of Ontologies in Support of Ontology Life Cycle" (2014-2016) which resulted in 25 publications.[2] I would like it to be noted that this was during the years in which I was working on the topic of this thesis with my colleagues. However, I consider the content of this thesis to be my proper research contribution. The initial idea of the OntoFarm collection is owing to my colleague, Vojtěch

[2] Publications are listed at `https://owl.vse.cz/COSOL/bibliography.html`

Svátek. However, I have been the maintainer of the collection from its inception.

The thesis begins with introductory Chapter 1, which presents semantic web ontology tools, their suggested categorization and the requirements for corresponding ontology benchmarks. Chapter 2 provides an overview of existing ontology benchmarks for different ontology tool categories. Moreover, it offers the rule-based recommendation system which is flexible in its knowledge base generation, based on the depicted information about ontology benchmarks available in Chapter 2. Chapter 3 provides details about the OntoFarm collection as an example of a manually constructed ontology benchmarking corpus. Chapter 4 describes the approach to an automatic support of ontology benchmarking corpus construction, implemented as the OOSP tool. Finally, Chapter 5 concludes the thesis by an introduction to the platforms enabling an automatic benchmarking of ontology tools.

I would like to thank my colleagues from the University of Economics, Prague as well as from universities and research institutions abroad. From the former, I am especially grateful to Vojtěch Svátek who opened the semantic web world to me and continuously provided me with invaluable research support. I would also like to thank my other colleagues from our "Semantic Web and Ontological Engineering" research group, especially Petr Strossa, Miroslav Vacura, Marek Dudáš and Daniel Vodňanský, from our "Intelligent Information Systems" research group, namely Petr Berka and from our "Data Mining and Knowledge Discovery" research group, namely Tomáš Kliegr. Further, I would like to thank my foreign colleagues Michelle Cheatham from Wrigh State University, US, Jerôme Euzenat and Jerôme David from INRIA, Grenoble, France, Ernesto Jiménez-Ruiz from the University of Oslo, Norway, Christian Meilicke and Heiner Stuckenschmidt from University of Mannheim, Germany, Cassia Trojahn from Université Toulousse, France and François Scharffe from Columbia University, US provided me an important feedback about using OntoFarm within the Ontology Alignment Evaluation Initiative context. Next, I would like to thank my wife Klára and my three sons Jan, Jeroným and Jáchym who support me in my life.

Ondřej Zamazal, August 31, 2018

Contents

Chapter 1

What is Benchmarking for Semantic Web?

In order to continuously enhance the quality of software, software engineering applies benchmarking as a method of measuring performance against a standard, or a given set of standards [75]. In contrast to software evaluation, benchmarking aims at continuous improvement with regard to a given set of standards known as benchmarks. Software evaluation and benchmarking are also important testing activities for semantic web tools. While the terminology about evaluation and benchmarking is not used uniquely within the semantic web, I will consider the evaluation as rather ad hoc software testing and benchmarking as a recurrent measuring activity related to the benchmark suite.

Although benchmarking activities and corresponding ontology benchmarks are common to semantic web field, they have not been uniformly characterized and summarized. In this chapter I provide a categorization of ontology tools along with typical requirements on ontology benchmarks for each ontology tool category and the next chapter provides a survey of existing ontology benchmarks within each ontology tool category along with their characteristics. In this respect the most relevant work was a dissertation thesis of García-Castro from 2009 [26] who focused on benchmarking methodology for semantic web technologies.

This habilitation thesis complements the work of García-Castro in a sense that I provide a summary of requirements of ontology tool categories on ontology benchmarks and I overview ontology benchmarks used in the semantic web using activities and characteristics from the proposed ontology tool categorization. Besides presented benchmarks ontology developers can also use freely available ontology collections (mentioned in Section 4.1). These are not included in the overview in Section 2 since they were not introduced as coherent benchmarks in literature before. The ontology collections can serve as a solid basis for constructing new ontology benchmarks as described in Chapter 4. Additionally, I provide the ontology tool benchmark recommender based on presented ontology benchmarks overview. This overview and instant recommendation aim at supporting researchers and ontology tool developers in their experimentation and benchmarking efforts.

1.1 Semantic Web and Semantic Web Ontology Tools

The most important and the most widespread service of the Internet, the World-Wide Web has evolved from textual static HTML pages to pages enriched with a multimedia content. The key term for WWW is *resource* which is anything that can be identified whether it is physical, abstract or digital thing. Semantic web represents one next (besides others) extension of WWW in terms of *semantics*. The main motivation for such an extension is to make available formal reasoning for more sophisticated software tools on WWW. This topic is covered by many books, e.g. [3], [2]. There are many use cases for semantic web applications such as product information aggregation from different product web pages where all of them use semantic web formats. The central role, for a representation at the semantic web, plays *ontologies*, sometimes also called *vocabularies*,[1] as formal conceptual models typically

[1] In this thesis I consider vocabularies as less expressive ontologies. The notion of ontology is further explained below in the text where OWL language is described.

describing a certain domain of discourse. In order to enhance building the semantic web applications there are many different ontology tools enabling realization of a full potential of semantic web technologies.

Semantic web is enabled by its architecture, the semantic web stack (Figure 1.1). This stack is fully rooted within the traditional web which is reflected by the traditional web technologies positioned at lower parts of the stack and dealing with a data transferring (HTTP), resource identification (IRI), character encoding (UNICODE) and data serialization (XML). Core semantic web technologies are placed above the traditional ones and they deal with data representation and interchange, *Resource Description Framework* (RDF).[2] Further, there is a semantically oriented *RDF Schema* (RDFS) [8] which enables us to construct simple ontologies by specifying classes of resources, relationships among resources using properties, domain and range of properties, taxonomy of classes and properties. This language can be extended by using the language constructs from *Web Ontology Language* (OWL) as explained in [40].

OWL is the most usual ontology language based on *description logics* (DL) as a family of logics being fragments of first order logic with useful computational features. Currently, the latest version is OWL 2 from December 2012. OWL language enables us to construct ontologies where the main building blocks are *entities* of different types: *individuals* referring to objects, e.g., Claire, *classes* referring to categories, e.g., Woman and *properties* referring to relationships, e.g., being spouse of somebody. Properties are of two types: *object properties* relating object to object, e.g., Claire to Andrew in the relationship of being a spouse, *datatype properties* attributing data value to object, e.g., a zip code to a city. OWL further allows to build complex description of concepts over entities by using *axioms* with OWL *constructs* such as existential restriction, e.g., an active student has at least one enrolled course. There are two kinds of axioms in ontology: *terminological* axioms (TBox) consisting of knowledge about entities in general and *assertional* axioms (ABox) describing particular objects from a

[2] `https://www.w3.org/TR/rdf-concepts/`

domain of interest. The OWL language has many different language constructs available for constructing axioms. The set of employed language constructs determines the complexity of the ontology. There are five basic profiles in OWL 2, as explained in [40], differing in their expressive power and corresponding computational costs. *OWL 2 EL* is mainly for large bio-health ontologies. *OWL 2 QL* is rather for simple ontologies such as a thesauri and such ontologies could be implemented in relational database technologies. *OWL 2 RL* is designed for applications requiring scalable reasoning. The profile *OWL 2 DL* corresponds to ontologies with the *SROIQ* description logic semantics. Ontologies beyond OWL 2 DL are considered as in *OWL 2 Full*.

Further, there are technologies for querying RDF data, e.g., *Simple Protocol and RDF Query Language* (SPARQL)[3] and for capturing rules, e.g., *Rule Interchange Format* (RIF)[4] beyond description logics. Proof and logics relate to the different technologies on layers below. For example, the primary purpose of developing ontologies was an option to infer an implicit taxonomy between classes and a categorization of individuals. This is realized by *ontology reasoners*.

Further layers, i.e., a trust and a cryptography, cope with technologies whose employment in the semantic web is still under a development and which should enhance a credibility to use of the semantic web applications. Each semantic web technology can be supported by corresponding semantic web tools, e.g., authoring RDFS or OWL ontologies, querying ontologies using SPARQL etc. In this chapter I focus on the semantic web tools which cope with semantic layers of the semantic web stack and particularly with ontologies, from now on shortly named as *ontology tools*. While an end-user interacts with a final semantic web application, ontology tools are rather intended for semantic web application developers.

The ontology has a crucial role for the semantic web. There are many different categories of tools supporting an ontology lifecycle. The central theme of this thesis is an ontology

[3] `https://www.w3.org/TR/sparql11-query/`

[4] `https://www.w3.org/2001/sw/wiki/RIF`

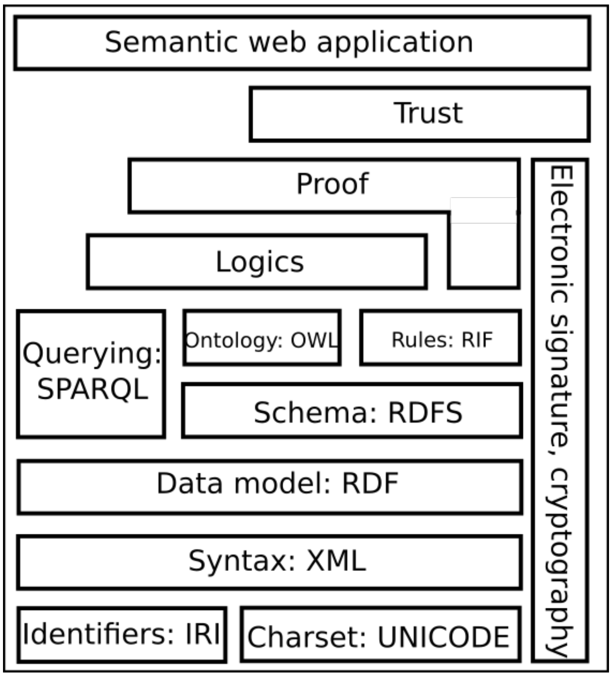

Figure 1.1: Semantic web stack. Source: [47]

benchmark which represents a material for benchmarking of ontology tools. In this thesis I provide a categorization of ontology tools which enables to describe each category using eight activities which can be performed on ontologies and each activity can be described using five characteristics. The relationships between categories, activities and characteristics are depicted on Figure 1.2. These concepts are explained in Section 1.1.1. Further, I provide typical requirements on ontology benchmarks for each ontology tool category in Section 1.2. Next, I provide a survey of existing ontology benchmarks grouped according to my ontology tool categorization and describe them using activities and characteristics from the categorization, Chapter 2. Further, according to this expert-based survey, I provide a simple recommender suggesting suitable ontology benchmarks according to a user's needs, Section 2.6. The recommender is based on rules, which are automatically induced from the ontology benchmarks characteristics stated in Chapter 2, and is highly modularized since a new set of rules can be automatically generated according to a given table of ontology benchmark characteristics on input. This recommendation is based on ontology benchmark descriptions using activities and their characteristics instead of predefined suitability of ontology benchmarks for some ontology tool category. The ontology tool categorization serves firstly for the grouping of ontology benchmarks and their description.

1.1.1 Ontology Tool Categories

Ontology tools can be categorized according to different aspects. Here I will categorize ontology tools according to their main purpose within an ontology lifecycle. According to authors in [66] there are many different activities supporting the ontology lifecycle. They cover *development*, *use* and *maintenance* phases. In the perspective of [66], the development is specified more generally as an ontology network development process (a feasibility study, an ontology specification etc). In this thesis I consider activities in the development phase as a support of creating a new ontology, activities in the use phase as an employment of an ontology in some application and activities in the maintenance phase as a continuous adjustment of an existing ontology. In this section, I will characterize several ontology tool categories (OTC). Figure 1.3 shows their position with regard to the ontology lifecycle phases.

Development of ontologies is supported by *ontology authoring tools* (OTC1). These tools enable users to design an ontology from scratch without reusing knowledge resources, e.g., Protégé [57]. Generally speaking, ontology authoring tools are capable to include other tools covering various other activities; typically by using plug-ins.

An ontology reflects knowledge from a certain domain of discourse. Since ontologies naturally reside in an open web space, their indirect (as an alignment) or direct interconnection (as imports) are commonly observed. This is supported by an ontology alignment where *ontology alignment tools* (OTC2), e.g., LogMap [43], serve as enablers to overcome the heterogeneity of ontologies from the same domain by providing *correspondences* between entities.

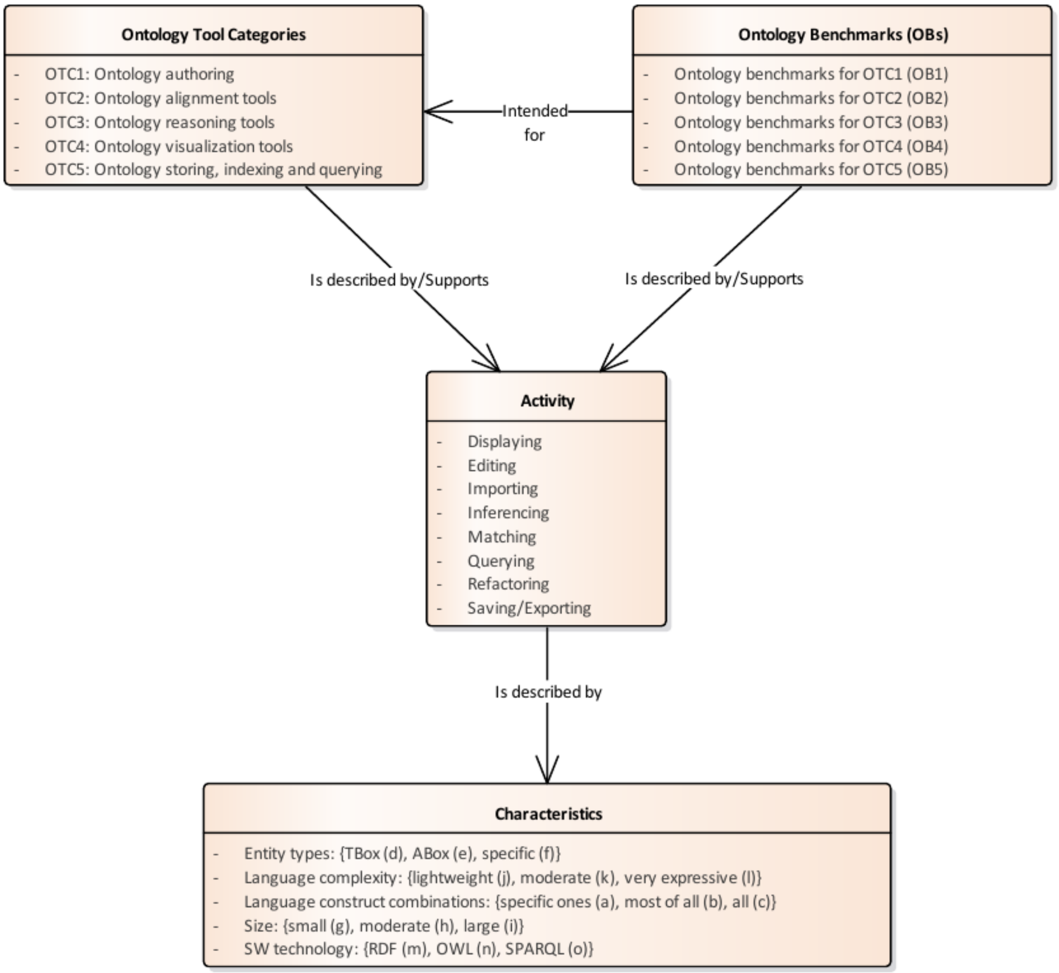
Ontology Tool Categories
- OTC1: Ontology authoring
- OTC2: Ontology alignment tools
- OTC3: Ontology reasoning tools
- OTC4: Ontology visualization tools
- OTC5: Ontology storing, indexing and querying
Intended for
Ontology Benchmarks (OBs)
- Ontology benchmarks for OTC1 (OB1)
- Ontology benchmarks for OTC2 (OB2)
- Ontology benchmarks for OTC3 (OB3)
- Ontology benchmarks for OTC4 (OB4)
- Ontology benchmarks for OTC5 (OB5)
Is described by/Supports
Is described by/Supports
Activity
- Displaying
- Editing
- Importing
- Inferencing
- Matching
- Querying
- Refactoring
- Saving/Exporting
Is described by
Characteristics
- Entity types: {TBox (d), ABox (e), specific (f)}
- Language complexity: {lightweight (j), moderate (k), very expressive (l)}
- Language construct combinations: {specific ones (a), most of all (b), all (c)}
- Size: {small (g), moderate (h), large (i)}
- SW technology: {RDF (m), OWL (n), SPARQL (o)}

Figure 1.2: Ontology tool categories, ontology benchmarks, activities and their characteristics. Source: Author.

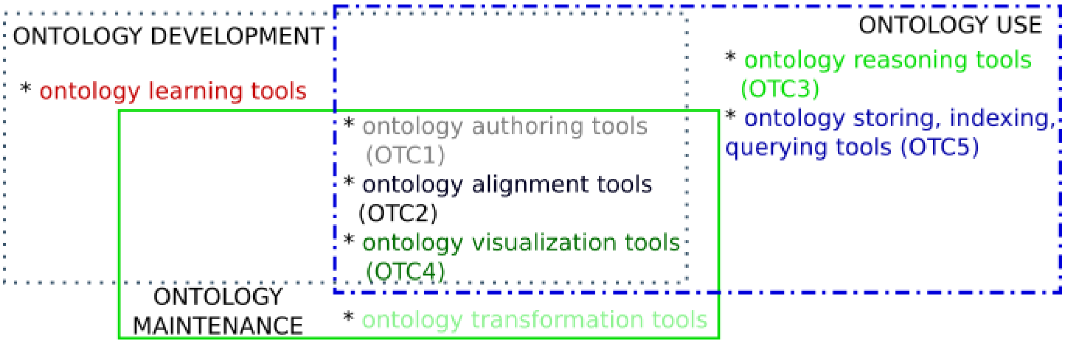
ONTOLOGY DEVELOPMENT
* ontology learning tools
ONTOLOGY USE
* ontology reasoning tools (OTC3)
* ontology storing, indexing, querying tools (OTC5)
* ontology authoring tools (OTC1)
* ontology alignment tools (OTC2)
* ontology visualization tools (OTC4)
ONTOLOGY MAINTENANCE
* ontology transformation tools

Figure 1.3: Ontology phases and ontology tool categories. Source: Author.

One of the original goals of the semantic web has been infer the knowledge using ontologies. *Ontology reasoning tools* (OTC3) enable to perform different reasoning tasks, e.g., entailment and consistency checking, over an ontology, e.g., HermiT [29]. During the ontology development, but also during common ontology use phase, the ontology users find it useful to visualize an ontology. There are many different *ontology visualization tools* (OTC4) that enable to perform various visualization tasks, e.g., VOWL [51].

Another important goal of the semantic web is to use ontologies for annotating data, in particular RDF data, to allow a semantic querying of the content. *Ontology storing, indexing and querying* (OTC5) includes tools which enable storing RDF data and querying the data captured according to an ontology using SPARQL, e.g., RDF4J (formerly known as Sesame) [9].

This ontology tool categorization is not complete since I only consider ontology tool categories for which there are some existing ontology benchmarks. For example, in this work I do not consider semi-automatic learning of ontologies, i.e., *ontology learning tools*, e.g., DL-Learner [10], belonging to the ontology development phase. Further, I omit *ontology transformation tools*, e.g., PatOMat framework [81], aiming at a transformation of lexical or structural aspect of an ontology during an ontology maintenance phase because it can happen that the knowledge is changed or that a designer of the ontology changed a perspective and thus ontology should be transformed accordingly.

In my ontology tool categorization I consider eight activities which are typically done with ontologies as artefacts:

- *Editing* activity aims at creating a new content or changing an existing content.
- *Importing* activity means loading an existing content into some artefact as its new part.
- *Saving/exporting* activity targets at serializing the content from the tool to some external output.
- *Displaying* activity deals with the visualization of some artefact or its part.
- *Inferencing* activity is about reasoning new facts or knowledge from existing facts and knowledge.
- *Refactoring* activity targets at changing an existing content or its part to improve the quality of the artefact.
- *Matching* activity is about searching for semantic connections between two or more artefacts.
- *Querying* activity aims at retrieving a certain part of some artefact based on some specification (query).

These activities are typically employed by ontology tools from different ontology tool categories, e.g., many different tool categories need to display an ontology or its part. However, different ontology tool categories need to perform an activity at various levels, e.g., displaying of an ontology in detail or displaying the whole ontology. To characterize an activity for each ontology tool category (or later for each ontology benchmark), I selected different specific characteristics which enable me to distinguish ontology tool categories (or ontology benchmarks):[5]

- *Language Construct Combinations* is about which combinations of ontology language constructs are covered: *specific* ones (a), which means a certain set of language constructs, e.g., class subsumption (rdfs:subclassof), class intersection (owl:intersectionOf),[6] *most* of all (b) which means many different language constructs from the given language, *all* (c) which means all language constructs of the given language.
- *Entity Types* target at which entity types are considered by an activity: entities from *TBox* (d), i.e., classes, properties, entities from *ABox* (e), i.e., individuals, or other *specific* (f)[7] elements.
- *Size* means how large is the artefact: *small* (g) which roughly means tens of entities, *moderate* (h) which roughly means hundreds of entities or *large* (i) which roughly means over one thousand of entities.
- *Language Complexity* is about an expressivity of given language behind: *lightweight* (j) which roughly means RDFS like expressivity (subsumptions, domains, ranges), *moderate* (k) which roughly means expressivity of OWL 2 EL (e.g., class disjointness, enumerations with a single individual, intersection of classes) or *very expressive* (l) which roughly means expressivity higher than OWL 2 EL.
- *SW Technology* deals with the prominent SW technology for an activity: *RDF* (m), *OWL* (n) or *SPARQL* (o).

[5] For each characteristic, which is explained below, there is the single-letter code in brackets for writing of characteristics in tables of this chapter, e.g., the *c* stands for all ontology language constructs.

[6] The prefix rdfs stands for `http://www.w3.org/2000/01/rdf-schema#` and the prefix owl stands for `http://www.w3.org/2002/07/owl#`. For a complete overview of ontology constructs for given language there are available corresponding specifications: [40] for OWL 2 and [8] for RDFS.

[7] Currently, this option is not used, however this was left in the categorization for some use in future, e.g., considering mappings as elements (e.g., instead of ontology classes) of benchmarking corpus.

Table 1.1: Ontology Tool Categories (OTC) and Their Typical Covered Activities. Source: Author.

Activity	OTC1	OTC2	OTC3	OTC4	OTC5
Editing	✓	✓			
Importing	✓	✓	✓	✓	✓
Saving/Exporting	✓	✓	✓	✓	✓
Displaying	✓	✓		✓	
Inferencing	✓	✓	✓		✓
Refactoring	✓				
Matching		✓			
Querying					✓

Each ontology tool category is described using the set of activities with their characteristics in Table 1.2.[8] Table 1.1 summarizes activities covered by ontology tool categories.

Based on this characterization we can see what each ontology tool category requires to be benchmarked for as described in Section 1.2. In order to find a match between ontology tool and ontology benchmark I characterize the ontology benchmarks using the same set of activities and their characteristics, Chapter 2. As a result, ontology benchmarks are recommended according to their supported activities and their characteristics. I conclude Chapter 2 by providing a recommender based on the gathered data from the characterization of the ontology benchmarks in Section 2.6.

[8] Characteristics of respective ontology tool categories are estimated from my own experience of working with ontology tools last 15 years. Since ontology tools within the ontology tool categories are naturally diverse in their features, stated typical characteristics in Table 1.2 are simplified to a certain extent.

Table 1.2: Typical Covered Activities of Tool Categories and Their Characteristics. Language construct combinations: specific ones (a), most of all (b), all (c). Entity types: TBox (d), ABox (e) or other specific (f) entity types. Size: small (g), moderate (h) or large (i). Language complexity: lightweight (j), moderate (k) or very expressive (l). SW technology: RDF (m), OWL (n) or SPARQL (o). Characteristics are described in Section 1.1.1. Source: Author.

	OTC1					OTC2					OTC3					OTC4					OTC5				
Activity	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology
Editing	c	d e	h	l	m n	a	d e	h i	k l	m n															
Importing	c	d e	h	l	m n	b	d e	h i	k l	m n	c	d e	h	l	n	c	d e	h	k	m n	c	d e	g h	k	m n o
Saving/ Exporting	c	d e	h	l	m n	b	d e	h i	k l	m n	c	d e	h	l	n	a	d e	h	k	n	c	d e	g h	k	m n o
Displaying	c	d e	h	k	n	a	d	h	k	n						c	d e	h i	k	m n					
Inferencing	b	d e	h	k	m n	a	d e	h	l	n	c	d e	h i	l	n						a	d e	g h	k	m n
Refactoring	a	d e	g	j	n																				
Matching						b	d e	h i	k l	m n															
Querying																					b	d e	g h	l	m n o

1.2 Requirements for Ontology Tool Benchmarks

In this section I will describe each ontology tool category with regard to their requirements for corresponding ontology benchmarks. Ontology authoring tools (OTC1) support an ontology designer during the ontology development process. Either, it means an ontology development from scratch or reuse of different knowledge resource to a various extent. One typical benchmarking case is to assess the interoperability and the coverage of various language combinations (Scenario 1).[9] This could involve an ontology on input or on output depending on which task (import/export) is benchmarked. Another way of benchmarking of this category of ontology tools includes an assignment of ontology conceptualization task for a user in a free text. This would involve a reference ontology on input.

Ontology learning includes a set of tasks and methods for the automatic or semi-automatic generation of ontologies from a natural language text as described in [65]. There can be various learning tasks such as concept formation, learning OWL class expressions and ontology population. Although each learning task has its own specifics, in general ontology learning tools work with a natural language text on input and output ontologies of different axiomatization (ranging from concept hierarchy, relations to non-trivial concept expressions). To the best of my knowledge there is no benchmark dealing with this ontology tool category.

Generally, benchmarking of ontology alignment (matching) tools (OTC2) requires pairs of ontologies from the same domain and a reference alignment between them (Scenario 2). Ontologies for the ontology alignment task used to be heterogeneous, which means that they differ lexicographically, structurally and/or semantically between each other provided they capture the same domain of discourse. A typical benchmark for ontology matching does not contain an ABox data set (individuals); however, if we consider an instance-matching task as a specific variant of ontology alignment then the ABox data set must be involved in the benchmark as well. Ontology alignment benchmark types mainly vary along the ontology characteristics. This means that for a specific ontology matching technique we can consider

[9] Solutions for four scenarios mentioned throughout this section are stated in Section 4.3.

corresponding specific ontologies, e.g., large ontologies (scalability issue), ontologies with ABox, structurally rich ontologies, ontologies rich in OWL language constructs, etc.

Benchmarking of ontology reasoning tools (OTC3) typically requires an ontology having non-trivial concept expressions (Scenario 3). The type of concept expressions depends on the reasoning task to be benchmarked, e.g., subsumptions, equivalences. If the tool needs benchmarking of reasoning tasks for the Abox, the benchmark should also involve an appropriate ABox data set described using the ontology from the benchmark. Often the ontology reasoners aim at the scalability issue (i.e., moderate or large ontologies).

Ontology visualization tools (OTC4) typically need to be benchmarked on whether they cover all language constructs of the OWL language (Scenario 4). The large size of an ontology could test the scalability of an implemented visualization technique.

Ontology storing, indexing and SPARQL querying tools (OTC5) naturally need to be benchmarked with regard to a storage performance against various ontologies and corresponding RDF data sets. It typically needs to be benchmarked for an answer querying over the data described according to a certain ontology. In this case, besides an ontology and a data set there must be some SPARQL queries. SPARQL queries often target at different combinations of SPARQL query language which correspond to different variations of the ontology language.

Generally, benchmarking of ontology transformation needs an ontology to be changed on input and corresponding reference ontology. Next, on input there is also expected a specification saying what should be changed and how. To the best of my knowledge there is no benchmark dealing with this ontology tool category.

The abovementioned requirements for the ontology tool benchmarks also reflect characteristics of the ontology tool categories depicted in Table 1.2. Figure 1.4 shows a position of the ontology benchmark groups with regard to the core semantic web technologies.

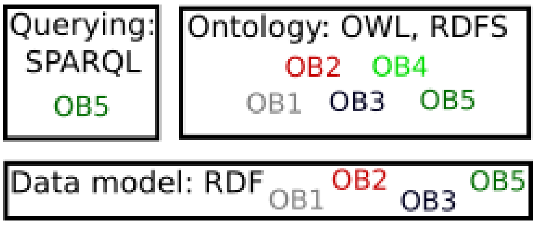

Figure 1.4: Ontology benchmarks position with regard to the core semantic web technologies from the semantic web stack where OB means "ontology benchmark". For numbering of ontology benchmark groups the same numbering as for ontology tool categories is applied, i.e., ontology benchmarks for ontology authoring tools (1), ontology benchmarks for ontology alignment tools (2), ontology benchmarks for reasoning tools (3), ontology benchmarks for visualization tools (4) and ontology benchmarks for ontology storing, indexing and SPARQL querying tools (5). Source: Author.

1.3 Ontology Tool Benchmark Construction

There are basically three approaches to ontology tool benchmark construction as depicted in Figure 1.5. They can be either manually or automatically constructed. The automatic approach can be of two types. Ontologies can be either *generated* based on required criteria or *searched for* in ontology repositories containing the existing ontologies. On the one hand, the generated ontologies usually comply with the required ontology characteristics. On the other hand, it is not always easy to specify those required criteria, and such ontologies are still synthetic. Thus, it is not always clear whether the support of such ontologies can be helpful in real scenarios. Searching and selecting existing ontologies makes the benchmark more realistic provided the ontology repositories do not contain only synthetic ontologies. Further, searching within the space of existing ontologies also enables us to apply alternative approaches in situations when a user does not know what to actually search for. In such a case, ontology similarity can assist in the construction of ontology benchmarks.

The benchmarks overviewed in Chapter 2 are assigned to those three approaches at the end of Chapter 2. Further, I present OntoFarm, manually constructed benchmark, in Chap-

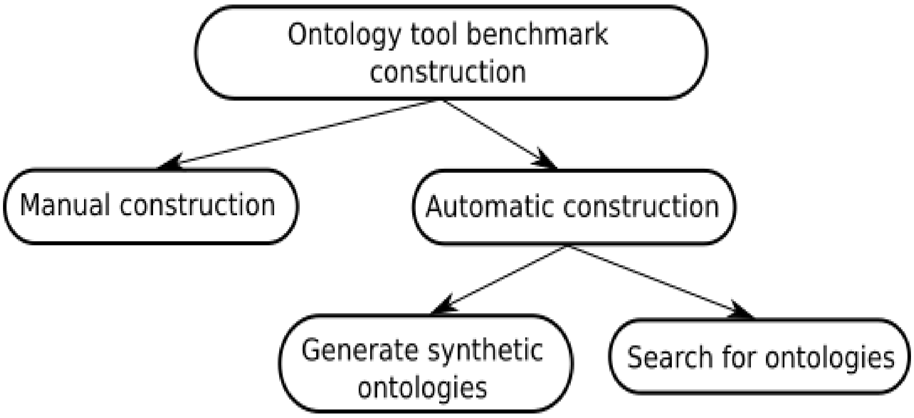
Ontology tool benchmark construction
Manual construction
Automatic construction
Generate synthetic ontologies
Search for ontologies

Figure 1.5: Ontology Tool Benchmark Construction. Source: Author.

ter 3 in more detail. In Chapter 4 I present my approach for supporting an automatic ontology tool benchmark construction. In Chapter 5 I provide an introduction to two platforms enabling an automatic benchmarking of ontology tools along with my experience from their usage.

Chapter 2

Ontology Benchmarks

This chapter contains an overview of existing ontology benchmarks for a different kind of ontology tool categories except ontology learning and ontology transformation for which there has not been any ontology benchmark prepared so far. If the benchmark data set or benchmark framework is not directly available (usually on the web), there is an asterisk stated next to its code name in corresponding table. The list with all links to considered benchmarks is online at `https://goo.gl/wut91K`. The chapter is concluded by proposed Ontology Benchmark Recommender.

The benchmarks are described in corresponding tables using activities and their characteristics according to my ontology tool categorization.[1] I further distinguish *minor* and *major* activities. While minor activities (depicted by normal text in the following tables) were not originally considered by authors of corresponding benchmark, activities which were considered by authors of corresponding benchmark are called major activities. These are depicted using bold text in the following tables. In comparison with major activities there are usually not explicit benchmarking metrics (measures) to use for minor activities in the

[1] Characteristics of respective benchmarks are estimated from their related literature and/or from my own experience with them. Since ontologies in ontology benchmarks are not always coherent in their features, corresponding characteristics might represent a certain approximation.

benchmark. However, for measuring the performance with regard to minor activities one can usually at least use basic approach inspecting whether a certain activity is supported within the benchmarked tool with regard to the ontology from given benchmark.

2.1 Ontology Authoring Tool Benchmarks (OB1)

The benchmarking of semantic web technologies was doctoral dissertation topic of Raúl García Castro presented in [26]. Castro mainly concentrated on a design of a benchmarking methodology for semantic web technologies and particularly on topic of interoperability. He proposed two interoperability benchmarks: the *RDF(S) Interoperability Benchmarking* that aims at interoperability using RDF(S) as the interchange language, and the *OWL Interoperability Benchmarking* that aims at interoperability using OWL as the interchange language.

The RDF(S) Benchmark Suite, **B1**,[2] evaluates the RDF(S) import and export functionalities of semantic web tools. The benchmark consists of several ontologies (benchmarks) serialized in an RDF/XML format. The evaluation aims at correctness of ontology import/-export and considers two modalities. The first modality checks an import/export correctness of all possible combinations of the model components. All possible combinations for import/-export are covered in three separated groups of benchmarks: importing/exporting ontologies with single components, importing/exporting ontologies with all possible combinations of two components, importing/exporting ontologies combining more than two components being usually together in RDF(S) models (e.g., *domain* and *range* of some *property*). In all, there can be over 4.000 benchmarks, however the benchmark suite can be reduced according to the kind of tools to be evaluated. Example of nine (ten) groups of benchmarks for import (export resp.) for ontology authoring tools is provided in [26]. The second modality

[2] I use codes *BX*, where X stands for a number, for referring to the ontology benchmarks in corresponding tables.

checks an import correctness of ontologies featuring different variants of RDF/XML syntax, e.g., there can be a different syntax for URI references, empty node abbreviations and string literal abbreviations. In the case of an export correctness of ontologies it deals with the component naming restriction where some characters are not allowed for representing RDF(S) or URIs, e.g., names with spaces.

The OWL Lite Benchmark, **B2**, only consists of an import part. Similarly to the RDF(S) benchmark it consists of ontologies aiming at correctness of importing with regard to different OWL vocabulary constructs. During the process of an ontology creation designers considered different possibilities of defining classes, properties and instances and used at most one or two OWL vocabulary constructs at the same time, while studied all the possible combinations with regard to other terms. Next, this benchmark also contains ontologies for benchmarking importing with regard to different variants of the RDF/XML syntax, e.g., URI references, abbreviations. The author of [26] concludes that this benchmark only targets at one aspect of interoperability issue, however there are further evaluation criteria that can be taken into consideration such as efficiency, scalability, robustness. Some of these other criteria are considered by benchmarks described below.

In 2004, W3C (the World Wide Web consortium) created a set of the RDF test cases, **B3**, presented in [31] and a set of the OWL test cases, **B4**, as presented in [11] and in [39]. These tests also include entailments and provide examples how to correctly use RDF and OWL and the formal meaning of their constructs. Therefore, these test cases can also be considered as interoperability benchmarks since they check the correctness of the tools with regard to dealing with RDF and OWL documents.

The characterization of four ontology authoring tool benchmarks is depicted in Table 2.1. On the one hand, with regard to the requirements for ontology benchmarks (Section 1.2) for ontology authoring tools these benchmarks correspond to benchmarks for interoperability (import/export) but there are no benchmarks regarding an ontology design based on a conceptualization described in a text. On the other hand, considering that there are collections of RDF or OWL documents covering different situations these benchmarks are also suitable, to a certain extent, for benchmarking editing, displaying and inferencing functionalities. These activities are thus depicted as minor ones except for inferencing which is major activity for B3 and B4.

2.2 Ontology Alignment Benchmarks (OB2)

Ontology alignment benchmarks aim at evaluating the performance of ontology matchers. In a nutshell, ontology matching targets at finding correspondences between semantically related entities of two ontologies. Since 2005 ontology alignment benchmarks started to be grouped around the *Ontology Alignment Evaluation Initiative* (OAEI) international campaign.[3] Originally, there were three tracks: the *anatomy* track dealing with a domain of body and consisting of two large ontologies (more than ten thousand classes), the *directory* track dealing with a domain of web sites directories and consisting of many test cases and the systematic *benchmark* track dealing with a domain of bibliography and one central ontology. In 2016, there were nine tracks. Here, I will describe nine tracks and their benchmarks from OAEI 2016 edition presented by [1] and one track from OAEI 2015 edition presented in [12]. Different tracks use benchmarks of a distinct nature and use different evaluation modalities. The most common evaluation modality is measuring the performance against a reference alignment. This performance is usually measured using *precision* (as the ratio of correctly found correspondences over all generated correspondences by the system), *recall* (as the ratio of correctly found correspondences over all expected correspondences by the reference alignment) and *F-measure* (as a harmonic mean of precision and recall). The characterization of ontology alignment benchmarks is depicted in Table 2.2 and in Table 2.3.

[3] In 2004 there were two forerunners connected to the Evaluation of Ontology-based Tools workshop and Information Interpretation and Integration Conference.

Table 2.1: Characteristics of Ontology Authoring Tools Benchmarks. Language construct combinations: specific ones (a), most of all (b), all (c). Entity types: TBox (d), ABox (e) or other specific (f) entity types. Size: small (g), moderate (h) or large (i). Language complexity: lightweight (j), moderate (k) or very expressive (l). SW technology: RDF (m), OWL (n) or SPARQL (o). Characteristics are described in Section 1.1.1. While inferencing is minor activity for B1 and B2, it is major activity for B3 and B4. Source: Author.

	B1					B2					B3					B4				
Activity	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology
Editing	c	d e	g	j	m	c	d e	g	k	n	c	d e	g	j	m	c	d e	g	l	n
Importing	c	d e	g	j	m	c	d e	g	k	n	c	d e	g	j	m	c	d e	g	l	n
Saving/ Exporting	c	d e	g	j	m						c	d e	g	j	m	c	d e	g	l	n
Displaying	c	d e	g	j	m	c	d e	g	k	n	c	d e	g	j	m	c	d e	g	l	n
Inferencing	c	d e	g	j	m	c	d e	g	k	n	**b**	**d** **e**	**g**	**j**	**m**	**b**	**d** **e**	**g**	**l**	**n**
Refactoring																				
Matching																				
Querying																				

2.2.1 Systematic Benchmarks (B5)

The goal of this systematic benchmark is to disclose strong or weak aspects of each matching algorithm by systematically altering a seed ontology. The traditional benchmark suite is generated based on the bibliographic ontology and in 2016 another benchmark was based on a film ontology. While the benchmark based on the bibliographic ontology was open, the benchmark based on the film ontology was not disclosed to the participants of the OAEI campaign.

Alterations of the seed ontology consist in discarding and modifying various ontology features such as names of entities, comments, the specialisation hierarchy, instances, properties and classes as described by [23]. Due to the synthetic nature of this benchmark it does not test matchers performance on real-life problems rather than it focuses on the detail characterization of the behaviour of the matchers.

2.2.2 Anatomy Benchmark (B6)

This benchmark consists of one test case dealing with matching between the adult mouse anatomy and a part of NCI thesaurus which describes the human anatomy. The goal of this track is to disclose matchers performance in a specific domain of biomedicine. There are usually large and carefully designed ontologies which are typically mainly based on technical terms without using a natural language. Further, those ontologies are used with specific annotations and the partOf relation.

While it is not difficult to find out trivial correspondences by simple string comparison techniques in this test case, for finding non-trivial correspondences it needs a specific analysis and often also a medical background knowledge. In 2017, current maintainers of the track summarized their experience in [19].

2.2.3 Conference Track and OntoFarm Benchmark (B7)

The OntoFarm collection, recently presented in [84], has been used within the conference track of OAEI since 2006. OntoFarm is a collection of heterogeneously structured ontologies describing the same domain of conference organization. Those ontologies are rich in OWL constructs and they are grounded in reality which means that their ontology designers developed the ontology according to either conference organization tool, web presenting the conference or personal experience of conference organizer. This grounding also enforced the heterogeneity.

The conference track contains reference alignment for seven ontologies, i.e., 21 reference alignments. Submitted alignments are evaluated using traditional precision and recall measures using reference alignment as well as its uncertain version, presented in [13], and using logical reasoning based on violations of conservativity principle, described in [64]. OntoFarm is presented in Chapter 3.

2.2.4 Large Biomedical Ontologies Benchmarks (B8)

Similarly to the anatomy track this track targets at biomedical ontologies as described by [42]. The specifics of this track consist in providing test cases with very large ontologies containing tens of thousands of classes: the foundational model of anatomy ontology (FMA), the systematic nomenclature of medicine ontology (SNOMED) and the national cancer institute thesaurus (NCI).

As the basis of the reference alignment was selected the UMLS metathesaurus as currently most comprehensive resource of integrating medical thesauri and ontologies, including FMA, SNOMED and NCI. It is known that reference alignments extracted from UMLS contain a significant number of logical inconsistencies. Due to the concerns about a fairness of removing correspondences causing incoherence, as described by [60], these correspondences were flagged in the latest benchmark and they are not considered neither as positive nor as negative during an evaluation. This should be fair for matchers not performing an automatic alignment repair as well as for matchers performing such a repair.

2.2.5 Disease and Phenotype Benchmarks (B9)

Similarly to the anatomy and the large biomedical ontologies tracks this track also aims at matching ontologies from biomedical domain as described in [37, 38]. In 2016, there were two test cases. The first test case consists in matching two disease ontologies and the second test case consists in matching two phenotype ontologies. There has not been a reference alignment, however the consensus alignments were generated based on alignments from participating matchers. This benchmark has been motivated by the fact that correspondences between these ontologies are currently curated manually by bioinformatics and disease experts. A matching automation would be beneficial for their work.

2.2.6 MultiFarm Benchmark (B10)

On the one side there was an increasing number of ontologies which did not use English language as a base language, on the other side there was no commonly accepted ontology matching benchmark targeting at a multilingualism. As a reaction to these two incentives the MultiFarm benchmark was introduced by [54] in 2012.

MultiFarm is composed of a set of seven ontologies, selected from the OntoFarm benchmark, for which a mutual reference alignment had been created manually. At the beginning the OntoFarm ontologies have been manually translated into eight languages other than English (Chinese, Czech, Dutch, French, German, Portuguese, Russian, and Spanish) and later on two next languages (Italian and Arabic) have been manually added. This manual translation assured its high quality. Each combination of ontologies and languages represents a test case for cross-lingual ontology matching leading to approximately 1500 test cases.

Table 2.2: Characteristics of Alignment Benchmarks Part I. Language construct combinations: specific ones (a), most of all (b), all (c). Entity types: TBox (d), ABox (e) or other specific (f) entity types. Size: small (g), moderate (h) or large (i). Language complexity: lightweight (j), moderate (k) or very expressive (l). SW technology: RDF (m), OWL (n) or SPARQL (o). Characteristics are described in Section 1.1.1. Source: Author.

Activity	B5					B6					B7					B8					B9				
	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology
Editing	b	d	h	k	n	b	d	i	k	n	b	d	h	l	n	b	d	i	k	n	b	d	i	k	n
Importing	b	d	h	k	n	b	d	i	k	n	b	d	h	l	n	b	d	i	k	n	b	d	i	k	n
Saving/ Exporting	b	d	h	k	n	b	d	i	k	n	b	d	h	l	n	b	d	i	k	n	b	d	i	k	n
Displaying	b	d	h	k	n	b	d	i	k	n	b	d	h	l	n	b	d	i	k	n	b	d	i	k	n
Inferencing	b	d	h	k	n	b	d	i	k	n	b	d	h	l	n	b	d	i	k	n	b	d	i	k	n
Refactoring																									
Matching	b	d	h	k	n	b	d	i	k	n	b	d	h	l	n	b	d	i	k	n	b	d	i	k	n
Querying																									
specifics	synthetic					biomedicine, BK need					conference domain					biomedicine, very large					biomedicine, very large				

2.2.7 Interactive Matching Evaluation Benchmarks (B11)

The goal of this track is to compare matchers which require a user interaction. It aims at showing if a user interaction can improve the matching results and how many interactions are necessary. This track does not provide its own benchmark data set, but it uses data sets from other OAEI tracks: anatomy, conference, large biomedical ontologies and disease and phenotype tracks. Its novelty consists in simulating an interactive matching, as described by [18].

Interactive matching is implemented via an oracle (based on a reference alignment) to which matchers can present a correspondence and the oracle tells whether the correspondence is correct or not. In order to simulate the possibility of user's errors by the oracle, there is an option to set an error probability where 0.0 means a perfect oracle.

2.2.8 Process Model Matching Benchmark (B12)

The goal of this track is to evaluate matchers on a specific task of matching process models. The benchmark contains nine process models representing the application process for a master program of German universities and reference alignments for all pairs of models. Originally, this benchmark has been introduced within the Process Model Matching Campaign 2015 introduced by [4].

Since process models were transformed from their original representation into a set of assertions (ABox) using the common ontology (TBox), the matching task is an instance matching task where a set of ABoxes share the same TBox. On contrary to ontologies, process models contain labels with verb-object phrases, e.g., *sending acceptance*, and a complex sequence of activities instead of a type hierarchy.

2.2.9 Instance Matching Benchmarks (B13)

In 2016 this track consisted of three benchmarks aiming at instance matching. The goal of the SABINE benchmark was to match instances of the class "Topic" where one ontology contains topics in English and another ontology contains topics in Italian. Additional task targets at discovering DBpedia entity best corresponding to each topic from the first ontology. Within the SYNTHETIC benchmark matchers should recognize when two instances, from two ontologies, describe the same instance (from domain of universities and creative works). This benchmark is generated synthetically. Finally, the DOREMUS benchmark contains two data sets coming from two French cultural institutions sharing one single ontology: the French national library and the Philharmonie de Paris. For the instance matching evaluation there are three modalities corresponding to different size of data sets and different degree of heterogeneity.

2.2.10 Ontology Alignment for Query Answering Benchmark (B14)

In 2014 and 2015 this benchmark, introduced in [17], targeted at an ontology based data access scenario where multiple ontologies can exist and it evaluates an ability of alignments to enable query answering. The benchmark consists of synthetic ABoxes for the ontologies from the OntoFarm collection. The scenario considers the situation (originally from the Optique project as described by [28]) where one ontology provides the vocabulary for formulating the queries and the second ontology is linked to the data and it is not visible to the users. The alignment enabling the integration represents one of the possible solutions.

Additionally, this benchmark aims at investigating the effects of violations of three different kinds of principles affecting the generated alignments as introducted by [44]: *consistency principle* means that the alignment should not lead to unsatisfiable classes in the integrated ontology, *locality principle* means that the correspondences should interconnect entities that have similar neighborhoods and *conservativity principle* means that the alignment should not introduce alterations in the classification of the input ontologies. Since 2016 this evaluation (except the locality principle) have become the part of the conference track of OAEI.

2.2.11 Other Benchmarks Used Within OAEI

Besides these ten suites of benchmarks from OAEI 2016 or 2015 there are also further benchmarks which were employed within older OAEI editions. I made their complete list available in the online table at `https://goo.gl/mdUxHY`. In all, there was 20 different matching tasks within the whole history of OAEI from 2004 till 2017, i.e., 15 editions including 2011.5 edition. The top seven tracks according to their frequency in editions of OAEI are as follows: 14 times systematic benchmarks[4] and the anatomy track, 13 times the conference track, nine times instance matching track, and seven times MultiFarm, large biomedical track and directory track. In early OAEI editions there were many tracks dealing with thesauri mapping such as the *directory* track aiming at mapping web directories such as Google, Yahoo or Looksmart; this TaxME benchmark is presented in [78]. Further, there were thesauri mapping tasks related to different domains such as food, environment, library and fisheries. All of them also aimed at multilinguality.

Since 2009 OAEI started with instance matching tracks. The main instance matching track was evolving since its inception in 2009. In 2009 authors in [24] introduced the *ISLab instance matching benchmark*. This benchmark[5] aimed at instance matching task and similarly as the benchmark track it was automatically generated using one data source being modified according to various criteria. It contained one small ontology and data about movies which were extracted from the IMDb web page.[6] The modifications applied on data were of three main categories: *value transformations* aimed at values changing in datatype properties. *Structural transformations* dealed with property values deletion, swapping datatype property to object property or separation one property value into more property values. Finally,

[4] This track was missing in the OAEI 2017.

[5] `http://islab.di.unimi.it/content/iimb2009/`

[6] `http://www.imdb.com/`

Table 2.3: Characteristics of Alignment Benchmarks Part II. Language construct combinations: specific ones (a), most of all (b), all (c). Entity types: TBox (d), ABox (e) or other specific (f) entity types. Size: small (g), moderate (h) or large (i). Language complexity: lightweight (j), moderate (k) or very expressive (l). SW technology: RDF (m), OWL (n) or SPARQL (o). Characteristics are described in Section 1.1.1. Source: Author.

Activity	B10					B11					B12					B13					B14				
	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology
Editing	b	d	h	l	n	b	d	h i	k l	n	b	d e	h	k	m n	b	d e	i	k	m n	b	d e	h	l	m n
Importing	b	d	h	l	n	b	d	h i	k l	n	b	d e	h	k	m n	b	d e	i	k	m n	b	d e	h	l	m n
Saving/ Exporting	b	d	h	l	n	b	d	h i	k l	n	b	d e	h	k	m n	b	d e	i	k	m n	b	d e	h	l	m n
Displaying	b	d	h	l	n	b	d	h i	k l	n	b	d e	h	k	m n	b	d e	i	k	m n	b	d e	h	l	m n
Inferencing	b	d	h	l	n	b	d	h i	k l	n	b	d e	h	k	m n	b	d e	i	k	m n	b	d e	h	l	m n
Refactoring																									
Matching	b	d	h	l	n	b	d	h i	k l	n	b	d e	h	k	m n	b	d e	i	k	m n	b	d e	h	l	m n
Querying																					a	d e	h	j	m n
specifics	multilinguality					interactivity					application process domain					partly synthetic					query answering				

there were *logical transformations* focusing on different instantiation of individuals referring to the same entity. This benchmark was involved in four consecutive OAEI editions starting in 2009. In 2010 the approach has been enhanced by using a collection of OWL ontologies and by taking a real-world data from the linked data cloud (the Freebase) on input applying the SWING (Semantic Web INstance Generation) approach as described in [25]. Besides this main track there were or still are further instance matching tracks such as *very large crosslingual track* targeted at large thesauri instance mapping and process model matching described in Section 2.2.8. In 2017 there was a new track, *HOBBIT Link Discovery*, dealing with instance matching specifically in a spatial domain.

We can conclude that the ontology alignment benchmarks are very matured and they correspond to specified ontology benchmarks requirements in Section 1.2. Table 2.2 and 2.3 indicate major activity characteristics for benchmarks (for OAEI 2016 and some for OAEI 2015) which could also be used for benchmarking of other activities (minor ones) despite its original focus on just matching. In this case I also state that those alignment benchmarks could be used for benchmarking editing, importing, exporting, displaying and inferencing activities for ontology tools. According to the stated activity characteristics the benchmarks are often indistinguishable, therefore I added a specific feature for each ontology alignment benchmark into the tables. In the case of B14 it could also be used for benchmarking querying. Since its range of language construct combinations and complexity in provided SPARQL queries is very limited, it is described as specific with regard to the language construct combinations and lightweight with regard to the complexity in Table 2.3.

2.3 Ontology Reasoning Tool Benchmarks (OB3) and SPARQL Benchmarks (OB5)

There are many traditional ontology benchmarks partly dealing with reasoning. One of such benchmarks, for reasoning, storage and querying, has been the Lehigh University Benchmark

(LUBM), **B15**, presented in [32]. It consists of an ontology for the university domain. LUBM allows to generate synthetic data sets scalable to an arbitrary size. Further, it contains 14 queries capturing various properties and several performance metrics, e.g., a load time, a query response time. Authors in [32] evaluated four OWL repositories (memory-based Sesame, OWLJessKB, database-based Sesame and DLDB-OWL). This benchmark does not aim at complex description logic reasoning.

While LUBM targets at evaluating the performance of OWL repositories regarding extensional queries over a large data set committing to a single realistic ontology, it does not aim at evaluating OWL repositories with respect to a certain given domain. To overcome this limitation authors in [74] suggested a new approach where synthetic data sets can be generated for any ontology. This generation is based on a probabilistic model which can generate synthetic data with similar properties as a representative data of a sample on input. This approach has been realized in a new benchmark the Lehigh BibTeX Benchmark (LBBM), **B16**, which was used for the same four OWL repositories as originally with LUBM. The different results showed the influence of ontology and data on performance of OWL repositories. Hence, it concludes that it is always needed to use a representative benchmark for the intended use case of OWL repository.

In 2006 Ma et al. proposed in [52] the University Ontology Benchmark (UOBM), **B17**, which extends LUBM in terms of inference and scalability issues. Original LUBM consists only a limited number of different kinds of OWL constructs (subset of OWL Lite) and is limited to its ability of measuring scalability of OWL repositories due to producing rather multiple isolated and small graphs instead of one hugely linked graph. UOBM contains OWL constructs of both OWL Lite and OWL DL profiles and the data generator also contains a property ensuring effective instance links generation to make synthesized benchmarks more realistic.

The above-mentioned benchmarks are closely related to SPARQL benchmarks where instead of an ontology there is rather a general logical schema. There are many famous

SPARQL benchmarks such as the Berlin SPARQL benchmark in [7] focusing on centralized querying. This benchmark is centered around an e-commerce use case where a set of products is offered by various vendors and different consumers have posted reviews about products. The benchmark defines an abstract data model for the use case along with two concrete representations: an RDF representation and a relational representation. The data about products are synthetically generated by the benchmark data generator according to some production rules.

Other benchmarks focus on a federated querying. First approach which considers federation at data level and does provide benchmarks consisting of multiple interlinked data sets is FedBench presented in [62]. FedBench contains real data sets (e.g., subset of DBpedia, NY Times, Drugbank) from the Linked Data cloud, multiple query sets and a comprehensive evaluation framework. The evaluation framework is a configurable Java benchmark driver and enables users to evaluate different scenarios. While this benchmark is customizable with regard to various use cases, it lacks scalability regarding the Linked Open cloud due to a few number of selected data sets and queries. In 2012 authors in [30] introduced a methodology for composing federated queries covering different characteristics and a toolkit for an automatic federated query generation, SPLODGE. The methodology first enables to characterize SPARQL queries to be generated using three sets of query characteristics: the query algebra (e.g., query type such as SELECT, CONSTRUCT), the query structure (e.g., different variable patterns) and the query cardinality (e.g., number of data sources involved in query answering).

Typical performance measures of SPARQL benchmarks are related to a processing time of queries such as *queries per second* or *overall runtime.*

Other benchmarks aim at evaluating the performance of just reasoners mostly measuring their *overall reasoning time*, *number of correct results* and *number of errors.* Authors in [58] aimed at reasoners using real-world OWL ontologies, **B18**. Based on the description of the benchmark in [58] I could not estimate what is the size of ontologies (authors wrote about a varying size). While Pan [58] focused on reasoning with whole ontologies, Bail et al. [5] presented a new approach, JustBench, focusing on a fine-grained benchmarking of how the reasoners and ontologies interact. JustBench, **B19**, is based on justifications for entailments of OWL ontologies. Thus, instead of evaluating the whole ontology individual justifications are evaluated for correctness and reasoner performance. This approach enables to design transparent analytic micro-benchmarks.

Further reasoning benchmarks merely focus on ABox reasoning. Authors in [76] discussed such benchmarks and suggested several new ones to support missing features. These benchmarks are out of the scope of this thesis.

There are also some approaches merely dealing with a benchmarking of description logic systems, e.g., [22] and [41] from 1998. While these approaches are not applicable for ontology benchmarking out-of-the-box, they can serve as a basis for some other ontology reasoning benchmarks.

To a certain extent RDF test cases [31] and OWL test cases [11] include reasoning benchmarking.

Since 2012 reasoners benchmarking have been shielded by an annual workshop OWL Reasoner Evaluation Workshop (ORE), **B20**. According to [59] in 2015 a data set used within this workshop contained 1920 ontologies sampled from three repositories: the BioPortal,[7] the Oxford Ontology Library,[8] and MOWLCorp [53] which is a corpus based on the 2014 snapshot of the web. The data set only contains ontologies having more than 50 axioms and the ontologies are approximated into OWL 2 DL profile. The data set is divided into several groups according to benchmarked reasoning tasks such as classification, consistency checking or instantiation.

The characterization of six reasoning benchmarks is depicted in Table 2.4. Although all of them are mostly focused on inferencing (and first three also on querying), they could

[7] `http://bioportal.bioontology.org/`

[8] `http://www.cs.ox.ac.uk/isg/ontologies/`

Table 2.4: Characteristics of SPARQL and Reasoning Benchmarks. Source: Author.

Activity	B15					B16*					B17					B18*					B19*					B20				
	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology	Combinations	Entity Types	Size	Complexity	SW Technology
Editing	a	d e	h	j	m n o	a	d e	g	j	m n o	b	d e	h	k	m n o	b	d e		j k l	n	b	d e	g	k	n	b	d e	g h i	j k l	n
Importing	a	d e	h	j	m n o	a	d e	g	j	m n o	b	d e	h	k	m n o	b	d e		j k l	n	b	d e	g	k	n	b	d e	g h i	j k l	n
Saving/ Exporting	a	d e	h	j	m n o	a	d e	g	j	m n o	b	d e	h	k	m n o	b	d e		j k l	n	b	d e	g	k	n	b	d e	g h i	j k l	n
Displaying	a	d e	h	j	m n o	a	d e	g	j	m n o	b	d e	h	k	m n o	b	d e		j k l	n	b	d e	g	k	n	b	d e	g h i	j k l	n
Inferencing	a	d e	h	j	m n o	a	d e	g	j	m n o	b	d e	h	k	m n o	b	d e		j k l	n	b	d e	g	k	n	b	d e	g h i	j k l	n
Refactoring																														
Matching																														
Querying	a	d e	h	j	m n o	a	d e	g	j	m n o	b	d e	h	k	m n o															

also be used for benchmarking editing, importing, exporting and displaying. The above-mentioned benchmarks include non-trivial concept expressions (corresponding well with reasoning benchmark requirement from Section 1.2) of different complexity.

2.4 Visualization Benchmarks (OB4)

In 2014 authors in [33] and [34] introduced a visualization benchmark, OntoViBe and OntoViBe2, **B21**, covering a wide variety of OWL 2 language constructs in order to enable the testing of ontology visualizations. This benchmark contains one ontology which has been designed to include a comprehensive set of OWL 2 language constructs and their systematic combinations. The ontology in the second version of the benchmark, OntoViBe 2, extends the first version by individuals, anonymous classes, annotations and different combinations of cardinality restrictions. On contrary to most other benchmarks, OntoViBe does not aim at the scalability, performance or efficiency issues, i.e., the number of entities in ontology (as one visualization benchmark requirement), but rather it aims at different supported features of tested systems and thus various combinations of language constructs (which is another visualization benchmark requirement). Regarding activities, covered by this benchmark, the main benchmarking activity is displaying where it has the following characterization $c|de|g|k|n$, i.e., language construct combinations(all), entity types(TBox, ABox), size(small), language complexity(moderate) and SW technology(OWL). It can also be used for benchmarking of editing, importing and exporting activities.

While OntoViBe provides the static ontology for benchmarking displaying, OntoBench presented in [50] enables user to generate OWL benchmark ontologies. Similarly to OntoViBe it also focuses on the OWL language construct coverage and concept combinations. A user can decide which OWL constructs should be included within a synthetic ontology. However, there are not any count-based ontology metrics metadata, e.g., number of leaves, maximal depth of the taxonomy, etc. There is also not any random feature in the OntoBench generator since for one combination of OWL constructs OntoBench always generates just one ontology. Although this could be useful when a user needs one ontology covering all combinations of concepts using a given set of OWL language constructs, it is rather clumsy in a situation where a user needs to get more ontologies with more or less variance. OntoBench is implemented as a web application and has already been validated in a specific case of visualization, where all OWL language constructs must be displayed properly. Synthetic ontologies by OntoBench do not reflect any specific domain of discourse and their naming is done in a way that is a self-explanatory (e.g., instance named as 'AllDifferent_Individual1') and as a consequence they are not meaningful for any real domain.

2.5 Ontology Benchmarks in Numbers

In theory there are 243 different combinations of benchmark characteristics. Table 2.5 reflect numbers of combinations for each activity benchmarks covered. There are distinguished major and minor activities. For each of them I counted number of all combinations (Comb.), number of unique combinations (Unique) and the ratio (or coverage) as a number of unique combinations to a number of all possible different combinations (Ratio). We can see that regarding major activities the inferencing activity has the highest ratio of unique different characteristic combinations; the ratio is equal to 0.17. The editing has not a coverage (the ratio is equal to zero) by any benchmark as a major activity. Regarding the coverage of minor activities, we can see that the editing and the displaying activities have the highest coverage of unique different characteristic combinations; the ratio is equal to 0.21. On contrary, the matching activity has not a coverage by any benchmark as minor at all. We can explain this by the fact that while the editing and the displaying are less demanding activities with regard to a preparation of benchmark, the matching activity requires not only ontologies featuring different characteristics but also a common domain of interest and ideally a reference alignment. From this perspective it is close to the querying activity

Table 2.5: Numbers of characteristic combinations in benchmarks for activities. Comb. means number of all combinations. Unique means number of all unique combinations and Ratio (coverage) means ratio of a number of unique combinations to a number of all possible different combinations. The highest ratios are in bold. The lowest ratios are in italic. Source: Author.

	Major			Minor			Major+Minor		
Activity	Comb.	Unique	Ratio	Comb.	Unique	Ratio	Comb.	Unique	Ratio
Editing	0	0	*0*	77	50	**0.21**	77	50	**0.21**
Importing	8	6	0.03	69	46	0.19	77	50	**0.21**
Exporting	6	4	0.02	69	46	0.19	75	50	**0.21**
Displaying	2	2	0.01	75	50	**0.21**	77	50	**0.21**
Inferencing	48	42	**0.17**	27	18	0.07	75	50	**0.21**
Matching	23	14	0.06	0	0	*0*	23	14	*0.06*
Querying	18	18	0.07	4	4	0.02	22	18	0.07

which requires another particularity: SPARQL queries. In all, the highest coverage of 0.21, corresponding to 50 different characteristic combinations, shows that there are many different characteristic combinations not covered by current benchmarks at all. It could be interesting to figure out whether some of those missing combinations are worth of considering during future preparation of new ontology benchmarks.

Table 2.6 shows numbers of benchmarks having certain characteristics. Thus, there can be maximally the number 21 corresponding to the number of all considered benchmarks. Benchmarks are divided according to their major and minor activities (major activities are in all capital letters). As we already mentioned, characteristic (f), which means specific for entity type, was not used however this was left there for some use in future, e.g., considering mappings as elements (e.g., instead of ontology classes) of benchmarking corpus.[9]

[9] I further omitted the option "unknown" which I used once with regard to the benchmark B18 for the size characterization.

Regarding the highest numbers within each characteristic group (such as language construct combinations, entity types etc.) and activity we can see that the most common for language construct combinations is most of all option except for querying where the most common is specific one option and except for major exporting, importing and displaying activities where the most common is the option all language combinations. This corresponds to specifically tailored benchmarks where all language combinations are present. In the case of entity types, the most common characteristic is TBox. While for all minor activities benchmarks rather feature moderate size ontologies, for major activities benchmarks most commonly have small ontologies except for matching where the most common is a moderate size. For the complexity, minor activities covered by benchmarks are mostly moderate. For major activities exporting, importing, inferencing, querying benchmarks are mostly of a lightweight complexity and for major activities displaying, matching benchmarks are mostly of a moderate complexity. Finally, the most common SW technology covered by benchmarks is naturally OWL except for the major exporting activity where the most common is RDF.

2.6 Ontology Benchmark Recommender

Ontology benchmark recommendation is a specific field where no (or very limited) data for previous recommendations is available. Therefore, for constructing benchmark recommendation system I use a rule-based system where the rules come from my expertise.

The *Ontology Benchmark Recommeder* (OBR),[10] Figure 2.1, is built as a knowledge base (KB) for the NEST expert system shell introduced by [6]. OBR is a web application which communicates with the NEST web service by sending the knowledge base and answers from an end-user and receiving results. NEST covers the functionality of non-compositional (Prolog-like) expert systems, traditional compositional rule-based expert systems (with un-

[10] The recommender is available at `http://owl.vse.cz/OBR/` and the knowledge base is available at `http://owl.vse.cz/KBforOBR/kb.xml`.

Table 2.6: Numbers of characteristics of benchmarks. Major activities are in all capital letters. Minor activities have all letters in lowercase but the first one. The highest numbers for each characteristic type and activity are in bold. Source: Author.

	Combinations			Entity Types			Size			Complexity			SW Technology		
Activity	specific ones (a)	most of all (b)	all (c)	TBox (d)	ABox (e)	specific (f)	small (g)	moderate (h)	large (i)	lightweight (j)	moderate (k)	very expressive (l)	RDF (m)	OWL (n)	SPARQL (o)
EDITING	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Editing	2	**14**	5	**21**	14	0	9	9	6	6	**13**	7	8	**19**	3
IMPORTING	0	0	**4**	4	4	0	**4**	0	0	**2**	1	1	2	2	0
Importing	2	**14**	1	**17**	10	0	5	**9**	6	4	**12**	6	6	**17**	3
EXPORTING	0	0	**3**	3	3	0	**3**	0	0	**2**	0	1	**2**	1	0
Exporting	2	**14**	1	**17**	10	0	5	**9**	6	4	**12**	6	6	**17**	3
DISPLAYING	0	0	**1**	1	1	0	**1**	0	0	0	**1**	0	0	**1**	0
Displaying	2	**14**	4	**20**	13	0	8	**9**	6	6	**12**	7	8	**18**	3
INFERENCING	2	**6**	0	8	8	0	**5**	3	1	**5**	4	3	4	**7**	3
Inferencing	0	**10**	2	**12**	5	0	3	**6**	5	1	**8**	4	4	**11**	0
MATCHING	0	**10**	0	**10**	3	0	1	**6**	5	0	**7**	4	3	**10**	0
Matching	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
QUERYING	**2**	1	0	3	3	0	1	**2**	0	**2**	1	0	3	3	3
Querying	**1**	0	0	1	1	0	0	**1**	0	**1**	0	0	1	1	0

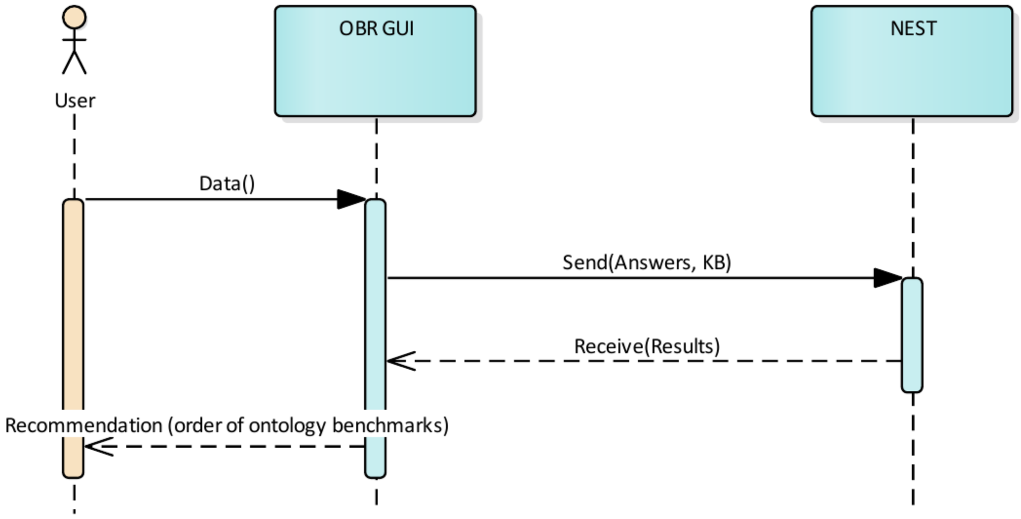

Figure 2.1: Ontology Benchmark Recommender. Source: Author.

certainty handling), and case-based reasoning systems. Representation of an uncertainty is realized in a standard range between -1 (certain FALSE) and +1 (certain TRUE) where 0 means irrelevant which have no effect on reasoning. For processing uncertainty NEST employs the algebraic theory of [35] and combination of backward and forward chaining. NEST offers the following building blocks to represent domain (task-specific) knowledge:

- *Attributes and propositions.* Characteristics of the consulted case are described using attributes. They have values from which propositions are derived. NEST employs four types of attributes: *binary* having just one corresponding proposition, *single nominal* having more than one proposition, *multiple nominal* having more than one proposion where any of them can be selected, and *numeric* having a number on input.

- *Rules* have a *condition* and *conclusion* where condition can have a disjunction of literal (as ‘attribute-value pair’) conjunctions and conclusion can have a list of literals. NEST employs three types of rules. In the case of *compositional* rule, its conclusion is bound to a weight. The weight captures the degree of uncertainty of the conclusion if the condition is certain. In order to assess the final weight of a proposition in the conclusion, all contributions of rules with this proposition in their conclusions must be combined. A compositional rule without condition is an *apriori* rule. Finally, there is a *logical* rule which is a non-compositional rule, i.e., without weights.

- *Contexts* are literals for determining that a rule can be applied. Contexts enable us to gather semantically related rules into semantically related parts of KB.

In order to support users I built the knowledge base (KB) for recommendation of ontology benchmark given users' preferences. The KB is automatically generated from tables depicting major and minor activities along with the characteristics of the benchmarks above. The whole code of the program in Scala for generating of KB is available at GitHub.[11] In all positive rules for inferencing benchmark recommendation I used as context (with 0.0 threshold) the activity for the benchmark:

```
activity : IF characteristic THEN benchmark [ weight ]
```

The setting of weights for rules is done using four meta-rules:

- using weight 0.7 if this is the characteristic of the benchmark in its major activity, e.g., CTXMatching : IF Size(h) THEN B10 [0.7]

- using weight 0.3 if this is the characteristic of the benchmark in its minor activity, e.g., CTXDisplaying : IF EntityTypes(d) THEN B9 [0.3]

- using weight -0.5 if this is not the characteristic of the benchmark in its major activity, e.g., CTXInferencing : IF LanguageComplexity(a) THEN B18 [-0.5]

- using weight -0.25 if this is not the characteristic of the benchmark in its minor activity, e.g., CTXImporting : IF EntityTypes(e) THEN B8 [-0.25]

[11] https://github.com/OndrejZamazal/KBGeneration/blob/master/GenerateKB.scala

Further, I generated negative logical rules (without context) with the threshold of 0.8 for benchmarks which do not support certain activity at all, e.g., IF matching THEN not B15. These rules are fired when given activity is very important for benchmarking, i.e., the weight of antecedent must be over 0.8.[12] The main advantage of such an automatic generation of KB is that this generation is easily tuned via meta-rules and easily reproducible via changed input tables.

In all, our KB contains seven attributes, 44 propositions, eight contexts, 1755 compositional rules with contexts (where 186 are positive rules for major activities, 249 are negative rules for major activities, 551 are positive rules for minor activities, 769 are negative rules for minor activities) and 51 negative logical rules. Five attributes ('Language Construct Combinations', 'Entity Types', 'Size', 'Language Complexity' and 'SW Technology') and their 15 propositions correspond to the characteristics as stated in Section 1.1.1. Next attribute ('Activity') and its eight propositions correspond to eight activities. Finally, one attribute ('Benchmark') corresponds to the set of 21 benchmarks. All attributes are multiple nominal which means that a user can select any number of propositions from corresponding attribute and set their weights, e.g., the size attribute has the following values (propositions): small, moderate or large.

2.6.1 Recommender Usage Example

In order to demostrate the recommender usage I include the example where a user needs to benchmark displaying while the most preferable is most language combinations option but having all of them is also a good option. Further, it should certainly be targeted at OWL with TBox having moderate or small size and moderate complexity. Lightweight ontologies are not preferred. Technically, the consultation consisted in answering the above-mentioned question with Likert-scale answers [49] (represented by numbers from interval -1;1 where 1/-

[12] All weights are set up based on my experimentation with trial consultations. However, further tuning would need to employ a validation which is left for future research.

1 means “Certainly yes”/“Certainly not”, 0.33/-0.33 means “perhaps yes”/“perhaps not” and 0.66/-0.66 means “probably yes”/“probably not”) as follows: Activity(Displaying: 1); LanguageElementCombinations(most: 1, all: 0.66); EntityTypes(TBox: 1, ABox: -1, specific: -1); size(moderate: 1, large: 0.33, small: 0.66); LanguageComplexity(lightweight: -1, moderate: 1, very expressive: 0.66); SWTechnology(OWL: 1, RDF: -1, SPARQL: -1). All other not mentioned propositions are irrelevant for inferencing, i.e., with weight 0. The recommendation (with weight 0.964), B21, can be considered as appropriate. The second most recommended benchmark is B20 (with weight 0.955) which nicely fits to the preferences of user and then B11 (with weight 0.909). Both, B20 and B11, actually comprise many diverse ontologies. B7 is recommended with weight 0.813 on the fourth position.

Let us consider one more consultation where a user needs to benchmark querying with preference of most or all language construct combinations. It should certainly be aimed at OWL with moderate or large TBox and lightweight complexity. In particular, this consultation contains the following answers: Activity(Querying: 1); LanguageElementCombinations(most: 1.0, all: 0.66; specific: -1); EntityTypes(TBox: 1, ABox: -1, specific: -1); size(moderate: 1.0, large: 0.66, small: -1); LanguageComplexity(lightweight: 1); SWTechnology(OWL: 1, RDF: -1, SPARQL: -1). The recommendation (with weight 0.977) is B15 and B17. Although they are not perfect matches, they can be considered as appropriate. While B15 has only not-preferred specific language construct combinations instead of preferred most of all combinations, B17 has a moderate complexity instead of preferred lightweight one. The third most recommended benchmark is B16 with weight 0.675. While the user preferred not small ontologies but moderate size of ontologies, B16 has small ontologies. Additionally, it also has not-preferred specific language construct combinations. B14 is recommended with weight 0.571 on the fourth position.[13]

[13] These and one more example consultations are available at the OBR web page.

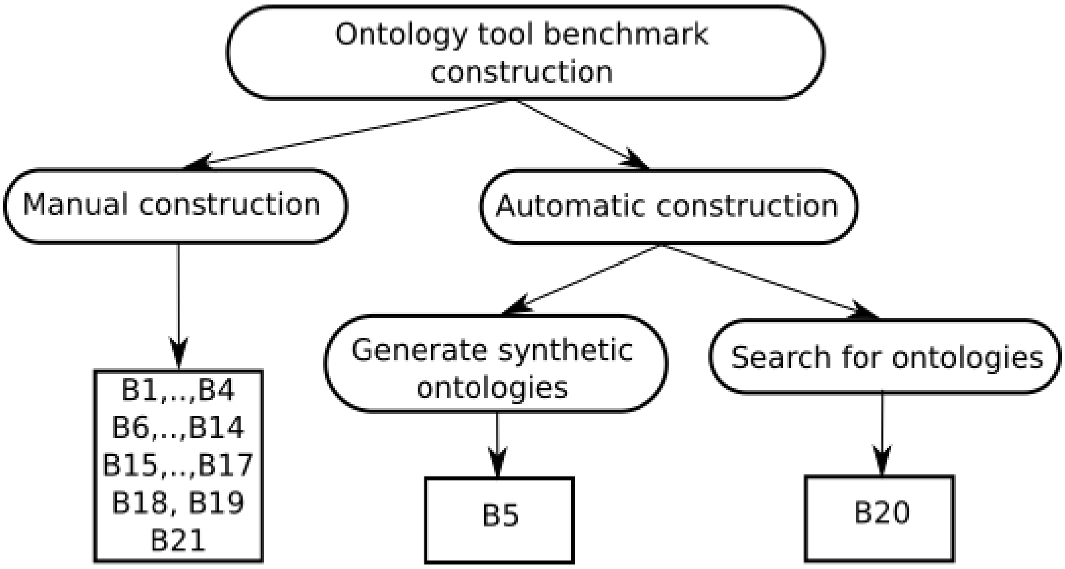

Figure 2.2: Ontology Tool Benchmark Construction and Particular Benchmarks. Source: Author.

2.7 Chapter Summary

This chapter provides the overview and the categorization of the semantic web benchmarks where an ontology plays an important role (ontology authoring, ontology alignment, ontology reasoning, ontology quering and ontology visualization). It is based on the semantic web ontology tool categorization introduced in Section 1.1.1. Although, there are many existing ontology benchmarks available, there has not been any available summary of them so far. This chapter can work as a navigator of a user to an adequate ontology benchmark.

We can categorize presented benchmarks according to the approach used for its construction as stated in Section 1.3 (the manual approach, the automatic by synthesizing ontologies approach, the automatic by searching for ontologies approach). The highest number of benchmarks are manually constructed. The benchmark B5 is generated using synthetic ontologies where one seed ontology is altered in different ways. The benchmarks B16 and B17 also have synthetic feature but this is applied on a generation of ABox data sets. Finally, the benchmark B20 is automatically constructed using ontology search as depicted in Figure 2.2.

According to the comparison between typical characteristics of Ontology Tool Categories (summarized in Table 1.2) and ontology benchmarks (Tables 2.1, 2.2, 2.3 and 2.4) we can conclude that the ontology benchmarks available for ontology authoring tools (B1..B4) only contain small artifacts and a refactoring activity is not covered at all. The ontology benchmarks for SPARQL querying (B15..B17) miss very expressive ontologies and all language construct combinations for querying activity. The ontology benchmarks for ontology reasoning tools (B18..B20) do not feature all language construct combinations. Next, the ontology benchmark for ontology visualization only contains rather small ontologies and do not feature RDF. Finally, the ontology benchmarks for ontology alignment can be considered as the most matured since they cover all typical characteristics of its field.

Moreover, I designed the rule-based system for recommendation of a suitable benchmark. The benchmark recommendation is a specific field where no (or very limited) data for previous recommendations is available. Hence, for building benchmark recommendation system I decided to use a rule-based system where the rules come from my expertise. The recommender is flexible in its knowledge base generation based on meta-rules and depicted information about ontology benchmarks described in this chapter. While the meta-rules can be considered as a certain approximation (they can be further tuned), it enables a rapid knowledge base generation. Further, the flexibility of the knowledge base generation is also related to possible extensions of the recommender by simply providing more ontology benchmarks and their characteristics into input table. In future the recommender could gather feedback from users of the recommender and this data could be used for more appropriate KB construction.

It should be noted that the recommendation is based on activities and their characteristics instead of some predefined ontology benchmark category. While the categorization of ontology benchmarks is useful for a primary navigation within the ontology benchmarks I think that the recommendation merely based on this could omit some suitable benchmarks, e.g., if ontology tool designer needs to benchmark displaying, s(h)e can use not only benchmarks for ontology visualization tools but also other benchmarks which support this activity to the required level.

Since our approach recommends an ontology benchmark based on required benchmarked activities, there could be an obvious extension in terms of a recommendation of ontology benchmark to given ontology tool category. To a certain extent, this can be already done using current recommender by applying the activities and their characteristics for given ontology tool category from Table 1.2. However, since Table 1.2 only contains typical characteristics of ontology tool categories, which are necessarily simplified, I consider using particular required activities and characteristics for given benchmarking need as more appropriate approach for making an ontology benchmark recommendation.

Chapter 3

OntoFarm: Manually Constructed Ontology Benchmark

This chapter provides details about the OntoFarm collection. The collection has been made manually on contrary to an automatic support for construction of benchmarking corpora which will be presented in Chapter 4.

The OntoFarm benchmark was introduced by Šváb et al. in 2005 in poster paper [67]. It contains a benchmark of heterogeneously structured ontologies describing the same domain of conference organization. Since 2006 it has been repeatedly used for benchmarking purposes within the OAEI 'conference' track, but also in several other projects. This chapter provides and overview of the benchmark and summary of its requirements, its history, its usage in projects and results of users survey. This chapter includes my updated journal article published in Journal of Web Semantics in 2017 [84].[1] I am the main author of the paper while my co-author, Vojtěch Svátek, iteratively provided me feedback to the content and edited the language.

[1] According to the permissions granted by the Elsevier publisher authors can include their articles in full or in part in a thesis or dissertation for non-commercial purposes as stated at `https://www.elsevier.com/about/our-business/policies/copyright/personal-use`

3.1 Requirements and Their Implementation

In 2005 due to the lack of experimentation ontologies we decided on several requirements for a benchmark of OWL ontologies suitable for experimentation with ontology tools:

- *Richness in OWL constructs.* OWL language provides to use various types of logical constructs for designing ontologies. In order to enable experimentation with ontology matchers focusing on different types of logical language constructs, it is important to include ontologies rich in this aspect. Obviously, this requirement is also relevant for experimentation with other types of ontology tools, e.g., reasoners and ontology visualization tools.

- *Sharing of the same, comprehensible domain.* In order to enable experimentation with ontology matchers, the ontologies must share the same domain (this requirement is specific to ontology matching). Moreover, it is important for the domain to be comprehensible since this boosts a fluent adoption by the community of ontology tool developers and researchers. Comprehensibility is a common requirement for experimentation benchmarks regardless the type of ontology tool (matcher, reasoner, visualizer or other).

- *Grounding in reality.* Connection to the reality further contributes to usefulness of an ontology benchmark. Due to this grounding on the one hand the ontology users can better understand the ontologies and on the other hand the ontology tools could exploit the data sources associated with the conceptualizations upon which the ontologies have been built. This requirement is also common for experimentation with many different types of ontology tools.

- *Natural presence of structural heterogeneity.* Structurally heterogeneous ontologies is the key challenge for ontology matching. In order to address this challenge it is required to naturally involve heterogeneity in the ontologies from the benchmark.

The OntoFarm benchmark reflects these requirements as follows:

- *Richness in OWL constructs* has been promoted by having the ontologies developed by people with at least minimal training in OWL (graduate students or researchers familiar with this field), including its more advanced TBox structures (going at least beyond RDFS).
- As *shared and comprehensible domain* we chose 'conference organization', as it is a familiar domain for all academic people. Moreover, this domain also shares some aspects with business activities, e.g., access restrictions to personal or sensitive data (such as reviews), hard vs. soft constraints, temporal dependencies among events, evolution of the meaning of concepts in time etc.
- OntoFarm ontologies have been *grounded in reality* by building each ontology upon one a real-world resource belonging to one of three different types: conference organization tool, web page of a conference, or personal experience from organizing a conference.
- OntoFarm ontologies *naturally feature heterogeneity* thanks to their abovementioned grounding on different real-world resources and their development by different people who did not have concrete guidelines and did not discuss among themselves.

We believe (and the survey results from Section 3.5 partly confirm it) that it was the fulfillment of these requirements that has led to solid uptake of OntoFarm by experimenters, despite its relatively tiny size and other limitations mentioned later.

3.2 History and Involvement in OAEI

Here I present a rough timeline of the OntoFarm versions and their use in OAEI.

- In *November 2005* the nucleus of OntoFarm comprising 4 ontologies has been presented within an ISWC 2005 poster paper [67]. Its main goal has been to provide a solid material for ontology experimentation according to the proposed requirements presented in Section 3.1. A simple demo application for ontology similarity computation has been presented along with the collection itself.

- In *November 2006* OntoFarm, already featuring ten ontologies, has been adopted by OAEI as one of its benchmarks.

- In *October 2008* the first 10 (pairwise) reference alignments for five ontologies (cmt, confOf, ekaw, iasted, sigkdd) were created.

- In *July 2009* the mutual alignments were built among all pairs of selected seven ontologies (i.e., 21 alignments).

- In *November 2011* the reference alignment was extended with further correspondences and the conflicting ones were resolved.

- In *2011* OntoFarm has been selected as the basis for generating a multilingual (eight-language) benchmark for ontology matching, called *MultiFarm*. MultiFarm is described in Section 2.2.6.

- In *August 2015* we prepared a variant of reference alignment that is not only free of inconsistencies but also free of violation of the conservativity principle (see Section 3.3.2 for more detail).

- In *February 2018* we started to work on complex reference alignment. This was successfully finished in May 2018. Now, it is a part of a new track in OAEI 2018 named as Complex track[2] as described in [69].

The OntoFarm ontologies as well as reference alignments are characterized in Section 3.3.

[2] `http://oaei.ontologymatching.org/2018/complex/index.html`

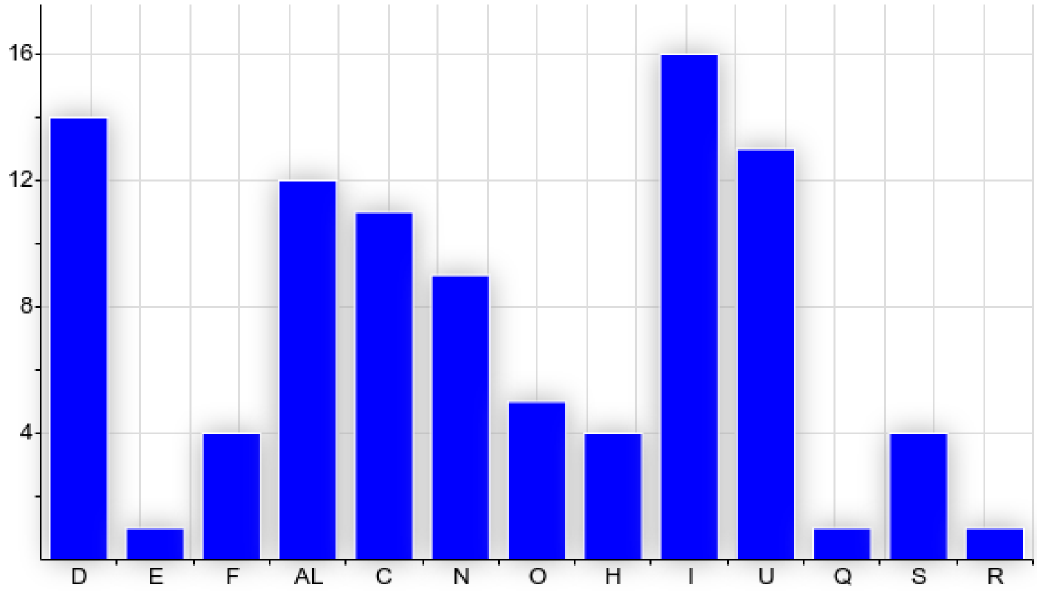

Figure 3.1: OntoFarm ontologies distribution according to DL constructs. D=Use of datatype properties, data values or data types. E=Full existential qualification. F=Functional properties. AL=Attributive language. C=Complex concept negation. N=Cardinality restrictions. O=Nominals (enumerated classes of object value restrictions - owl:oneOf, owl:hasValue). H=Role hierarchy (subproperties - rdfs:subPropertyOf). I=Inverse properties. U=Concept union. Q=Qualified cardinality restrictions. S=Attributive Language with Complements and transitive roles. R=Complex role inclusion. Source: [84].

3.3 Collection Overview

3.3.1 OntoFarm Ontologies

The OntoFarm benchmark contains 16 ontologies, which are small (tens to hundreds of axioms) but relatively rich in different language primitives. Figure 3.1[3] shows the number of OntoFarm ontologies employing a certain Description Logic (DL) construct, to give a rough idea on their expressiveness scope.

The list of all ontologies along with their basic metrics with selected details about en-

[3] The figure is automatically generated by my tool OOSP described in Chapter 4.

Table 3.1: Basic metrics for OntoFarm and selected details about entailments where inferred subclassof axioms do not include asserted ones; there is a number of inferred axioms without owl:Thing in a superclass position in brackets. The highest numbers are in bold. Different grounding in reality of ontologies is depicted by specific font used for an ontology name: ontologies based on personal experience from organizing a conference are in bold, ontologies based on the web of a conference are in italic, other ontologies are based on conference organization tool. Source: [84].

ontology	#classes	#data properties	#object properties	#individuals	classification time	#inferred subclassof
cmt	30	10	49	0	210	11 (4)
cocus	55	0	34	0	110	14 (8)
conference	60	18	46	0	27	11 (0)
confious	57	5	51	17	50	8 (0)
confOf	39	**23**	13	0	9	5 (0)
crs_dr	14	2	15	0	2	4 (0)
edas	104	20	30	**114**	20	**19** (4)
ekaw	74	0	33	0	10	6 (0)
iasted	**141**	3	38	4	**4414**	11 (3)
linklings	37	14	29	5	31	17 (6)
micro	32	9	17	4	10	3 (0)
myreview	39	17	49	2	23	5 (0)
openconf	62	21	24	7	27	16 (**14**)
paperdyne	46	20	**58**	0	19	6 (2)
pcs	24	14	24	0	4	6 (0)
sigkdd	50	11	17	0	7	8 (1)
mean	54	12	33	10	311	9 (3)

tailments are provided in Table 3.1. Classification time is generally very short (311 ms on average). The number of inferred and not asserted subclassof axioms is generally low ranging from 3 to 19. On average there are only three inferred subclassof axioms per ontology which do not have owl:Thing in superclass position. These can be considered as non-trivial entailments which are typically justified with a mutual interplay of class and/or property definitions.

According to the grounding of the OntoFarm ontologies in reality (for three different real-world resources see Section 3.1 and Table 3.1) we can point out some qualitative characteristics. Ontologies which are based on conference organization tools tend to be more technical (e.g., *Conference_setup* in *confious*, *Setup_Phase* in *paperdyne*). On contrary, ontologies based on a conference web usually put more emphasis on the payment aspect (e.g., *Fee* and its subclasses *Registration_fee* and *Sponzor_fee* in *sigkdd*, *Money* and its subclasses in *iasted*). Finally, ontologies based on personal experience tend to conceptualize submitted contribution in detail (e.g., *Submitted_contribution* and its four subclasses in *conference*, *Submitted_Paper* and its four subclasses in *ekaw*).

In order to further characterize OntoFarm by other aspects we selected a summary of typical characteristics from the ontology metrics provided by the "Online Ontology Set Picker" (OOSP) tool [80], see Table 3.2.[4] All OntoFarm ontologies contain a certain number of classes and object properties; a couple of them however do not have datatype properties, and more than half of the ontologies (9) do not include instances.

Classes are typically structured into 4 layers, however there is, on the one side, also an ontology having 7 layers, and, on the other side, an ontology only having 2 layers. There are large differences in the number of leaf classes among the ontologies (the standard deviation equals to 23). Although ontologies typically have many subclass axioms (78 on average, but also with a high standard deviation of 55), multiple inheritance is infrequent: there are only

[4] A snapshot of the ontology collection along with its statistics is available at `https://owl.vse.cz/ontofarm/statistics/`

Table 3.2: Selected summary statistics for the OntoFarm benchmark. Source: [84].

metric	#nonzero	min	mean	standard deviation	max
classes	16	14	54	31	141
object properties	16	13	33	14	58
datatype properties	14	0	12	8	23
instances	7	0	9.6	27	114
layers	16	2	4	1	7
top classes	16	4	9	5	20
leaf classes	16	11	40	23	103
subclasses	16	10	78	55	247
multiple inheritance	3	0	1	4	14
named domain	16	7	34	15	62
anonymous domain	13	0	4	3	11
named range	16	12	27	12	49
anonymous range	10	0	3	3	10

3 ontologies exhibiting it (however, one of them has 14 cases of multiple inheritance).

While using named classes as domain/range definition is typical for the OntoFarm ontologies (on average they have 34/27 of them, respectively), anonymous classes are only used infrequently for domain/range (on average 4/3, respectively) but still present in most ontologies (in 13/10, respectively).

3.3.2 OntoFarm Reference Alignments

In the course of time, three sets of reference alignments, based on seven of the OntoFarm ontologies, have gradually emerged. The original reference alignments from 2009, *ra1*, have been built by three evaluators who evaluated matches independently and then discussed the contradictory cases in order to arrive at consensus. The ontologies had been selected according to their size and quality based on the judgement of the experts involved in reference alignments construction. In 2011 the original reference alignment has been extended by computing a transitive closure and in order to obtain a coherent result, conflicting cor-

Table 3.3: The list of reference alignments for the three variants (ra1, ra2 and rar2). There are numbers of matches and ratios of classes and properties involved in matches from each ontology ($O_1|O_2$) per reference alignment. The highest values are in bold. The lowest values are in bold italic. Source: [84].

	ra1			ra2			rar2		
O_1-O_2	#matches	#classes	#prop.	#matches	#classes	#prop.	#matches	#classes	#prop.
Cmt-conference	15	.38|.20	.05|.05	17	**.48**|.22	.05|.05	17	**.48**|.22	.05|.05
Cmt-confof	16	.34|.26	.10|**.17**	14	.31|.24	.08|**.14**	13	.28|.21	.08|**.14**
Cmt-edas	13	.28|.08	.08|.10	15	.31|.09	.10|.12	15	.31|.09	.10|.12
Cmt-ekaw	11	.28|.11	.05|.09	13	.34|.14	.05|.09	12	.31|.12	.05|.09
Cmt-iasted	***4***	.14|***.03***	***.00|.00***	***4***	.14|***.03***	***.00|.00***	***4***	.14|***.03***	***.00|.00***
Cmt-sigkdd	12	.34|.20	.03|.07	13	.38|.22	.03|.07	13	.38|.22	.03|.07
Conference-confof	15	.18|.29	.06|.11	16	.20|.32	.06|.11	13	.17|.26	.05|.08
Conference-edas	17	.23|.14	.05|.06	17	.23|.14	.05|.06	17	.23|.14	.05|.06
Conference-ekaw	**25**	.38|**.32**	.03|.06	**26**	.40|**.33**	.03|.06	22	.33|.27	.03|.06
Conference-iasted	14	.22|.09	.02|.02	14	.22|.09	.02|.02	12	.18|.08	.02|.02
Conference-sigkdd	15	.20|.24	.05|.11	17	.23|.29	.05|.11	16	.22|.27	.05|.11
Confof-edas	19	.37|.14	**.14**|.10	19	.39|.15	**.11**|.08	19	.39|.15	**.11**|.08
Confof-ekaw	20	**.53**|.26	***.00|.00***	18	.47|.25	***.00|.00***	17	.45|.23	***.00|.00***
Confof-iasted	9	.24|.06	***.00|.00***	10	.26|.07	***.00|.00***	9	.24|.06	***.00|.00***
Confof-sigkdd	7	.16|.12	.03|.04	7	.16|.12	.03|.04	7	.16|.12	.03|.04
Edas-ekaw	23	.18|.26	.08|.12	25	.20|.29	.08|.12	**24**	.19|.27	.08|.12
Edas-iasted	19	.18|.14	***.00|.00***	17	.17|.12	***.00|.00***	17	.17|.12	***.00|.00***
Edas-sigkdd	15	***.11***|.22	.08|.14	13	***.09***|.18	.08|**.14**	13	***.09***|.18	.08|**.14**
Ekaw-iasted	10	.14|.07	***.00|.00***	13	.16|.09	.03|.02	13	.16|.09	.03|.02
Ekaw-sigkdd	11	.15|.22	***.00|.00***	11	.15|.22	***.00|.00***	10	.14|.20	***.00|.00***
Iasted-sigkdd	15	***.11***|.31	***.00|.00***	16	.11|**.33**	***.00|.00***	16	.11|**.33**	***.00|.00***
mean	14.52	.24|.18	.04|.06	15.0	.26|.19	.04|.06	14.24	.24|.17	.04|.06

respondences, i.e., those causing unsatisfiability, were manually inspected and removed by evaluators, whose work was eased with a reasoning-based tool [55]. The resulting reference alignments are labelled as *ra2*. In 2015, we detected violations of conservativity using the approach by Solimando et al. [63] and resolved them by an evaluator. The resulting reference alignments, featuring slightly higher correctness and completeness, are labelled as *rar2*. Whereas the old reference alignments (*ra1*) are available, the new reference alignments remain closed to make the matching task more difficult.

Table 3.3 provides the numbers of matches and ratios of classes and properties involved in matches from each ontology ($O_1|O_2$) per reference alignment. These numbers can possibly serve as guidance for experimenters who only want to adopt a subset of the collection. The number of matches ranges from 4 to 26. On average *cmt* has the lowest number of matches as the smallest ontology and *edas* has the highest number of matches (except *ra2* where *conference* has the highest number). While the highest coverage of classes being matched for one ontology is 53% of the *confOf* classes aligned with *ekaw* (*ra1*), the lowest coverage of classes being matched for one ontology is 3% of the *iasted* classes aligned with *cmt*. Further, *confOf* has the highest coverage of classes (31%) being matched in *ra1* on average and *iasted* has the lowest coverage of classes (8%).[5] The best matchable ontology, on average, is *ekaw* (28% of classes from other ontologies are matchable) and the worst matchable ontology is *cmt* (15%). For *ra2* and *rar2* *cmt* has the highest coverage of classes (around 32%) being matched on average and again *iasted* has the lowest coverage of classes (around 12%). For these two reference alignment sets the best matchable ontology, on average, is *conference* (around 21% of classes from other ontologies are matchable) and the worst matchable ontology is again *cmt* (around 15%). Regarding properties while *edas* has the highest coverage of properties and also it is the best matchable ontology in this respect (both around 8%), on average, for all reference alignments, *iasted* has the lowest coverage of properties and it is the worst matchable ontology in this respect (both around 0.7%).

In 2018, three ontologies from OntoFarm have been selected for a construction of a consensual complex reference alignment. Three evaluators constructed complex reference alignment individually and then there was an extensive discussion to arrive at the consensual complex reference alignment where we proceeded according to the methodology described

[5] This part of the analysis is based on Table 3.3, however due to space the additional computations are swapped to `https://owl.vse.cz/ontofarm/ra.html`

in [71].

3.4 OntoFarm in Action

3.4.1 Collecting Papers Dealing with OntoFarm

For collecting items for the literature review I had to consider two specifics related to citing the OntoFarm benchmark within research papers. First, some research papers directly cite the OntoFarm ISWC poster paper from 2005 (Group 1). This poster paper is not indexed by the *Web of Science* database,[6] therefore I could not take advantage of its citation analysis. Second, the OntoFarm benchmark is best known via its use for the conference track of OAEI. As a consequence it is rather known as the *conference dataset* and research papers often cite one of the OAEI summary papers (Group 2). Considering these two specifics for referencing research papers, I used the *Google Scholar* service[7] to collect papers from those two groups. Further, I made these two groups disjoint in the sense that if some paper cited both the OntoFarm and OAEI papers then it was only included into Group 1. After collecting the referencing papers, I manually analyzed them to figure out whether they really deal with ontologi/es from the OntoFarm benchmark or only cite the paper as relevant literature. Only in the former case I counted it among the *papers using OntoFarm or being about OntoFarm*. In the case of OAEI (Group 2) the chance that a paper referencing the OAEI summary paper used ontologies from the OntoFarm benchmark was obviously much lower than in the case of papers referencing the OntoFarm paper (Group 1).

In all, I collected 41 papers from Group 1 as papers using OntoFarm or being about OntoFarm (I call it Group A) and, similarly, 41 papers from Group 2 (I call it Group B). Groups A and B include workshop papers, conference papers, book chapters and journal papers. In all, I gathered 82 relevant papers, out of which 6 contain the respective reference

[6] `http://isiknowledge.com/`

[7] `https://scholar.google.com/` updated on 06-06-18.

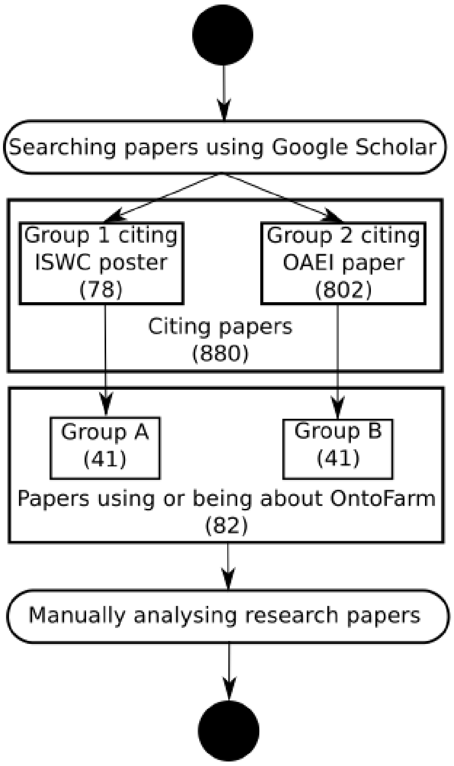
Searching papers using Google Scholar
Group 1 citing ISWC poster (78)
Group 2 citing OAEI paper (802)
Citing papers (880)
Group A (41)
Group B (41)
Papers using or being about OntoFarm (82)
Manually analysing research papers

Figure 3.2: Collecting relevant papers for literature review. There are numbers of respective papers in brackets. Source: Author.

as self-citation. It should be noted that many papers use OntoFarm although they cite neither the ISWC 2005 poster paper nor any OAEI summary paper; they instead often refer to the OntoFarm or conference track web page. Since such information is not indexed by publication databases, I intentionally ignored these papers in the literature review. The process of collecting relevant papers for literature review is depicted at Figure 3.2.

3.4.2 Literature Review

Table 3.4 presents the numbers of papers in each paper category.[8] Surprisingly, there are more conference papers than workshop papers. The distribution of papers according to paper

[8] All papers are listed in the OntoFarm web page.

categories is indifferent for Group A and Group B.

From all 82 papers I identified five papers that refer, in a sense, to derivatives of OntoFarm. In 2012 a group of researchers (including us) established a multilingual dataset based on OntoFarm, targeting multilingual ontology matching as presented by Meilicke et al. in [54]. Within the international cooperation we translated the OntoFarm ontologies into Czech language and we also collaborated on the initial inception of the MultiFarm benchmark and on initial evaluations as described by Meilicke in [56]. One of the latest contributions to MultiFarm was described in a paper written by Khiat et al. [48]. Further, Cheatham and Hitzler [13] prepared a version of reference alignment including the degree of agreement of experts' opinions on correspondences. This was first used within OAEI in 2015. In 2017 and 2018, the complex reference alignment built on the part of OntoFarm appeared in [70, 71].

We analyzed all 82 papers according to their prevalent *semantic web subfield* to which they belong.[9] The most common use of OntoFarm reported in papers is naturally related to evaluating *ontology matching techniques* (31 papers) where all but five used OntoFarm along with its reference alignments. *Ontology matching evaluation* includes 17 papers[10] where the authors propose a new evaluation approach, a new evaluation corpus or provide an evaluation of further alignment techniques. All but two papers used reference alignments. From its inception OntoFarm has been popular among researchers working in the *ontology matching debugging and repair* subfield, where I identified 14 papers. While those three subfields can be considered as core ontology matching subfields, OntoFarm has been also used in other ontology matching subfields to which less attention has been paid by researchers. OntoFarm was used in papers dealing with *complex correspondences and patterns* (5 papers) and provided working examples in *correspondence representation* papers (2). Five papers dealing with *application scenarios of ontology matching* exemplified them on OntoFarm ontologies. Despite the strong majority of papers dealing with ontology matching subfields, there are

[9] The list of papers according to their semantic web subfield is at `https://owl.vse.cz/ontofarm/swsubfields.html`

[10] There are also included the five mention OntoFarm-derivative papers.

Table 3.4: Relevant papers according to paper categories: W means workshop paper, C means conference paper, B means chapter in book and J means journal paper. Source: Author.

	Group A					Group B			
Year	W	C	B	J	$\sum$	W	C	J	$\sum$
2006	2	-	-	-	2	-	-	-	0
2007	1	2	-	-	3	2	1	-	3
2008	1	4	-	-	5	-	-	-	0
2009	3	2	1	1	7	-	1	2	3
2010	1	3	-	2	6	2	1	-	3
2011	1	2	-	1	4	1	4	-	5
2012	2	2	-	1	5	3	3	-	6
2013	-	-	-	-	-	-	3	3	6
2014	1	2	-	-	3	3	3	3	9
2015	1	1	-	-	2	2	1	2	5
2016	-	-	-	2	2	-	-	1	1
2017	1	-	-	-	-	-	-	-	1
2018	-	1	-	-	-	-	-	-	1
$\sum$	14	19	1	7	41	13	17	11	41

Table 3.5: Citing papers according to categories; *comp.* means complex, *corr.* means correspondence(s) and *pat.* means patterns. Source: Author.

SW subfield	W	C	B	J	$\sum$
OM technique	7	18	-	6	31
OM evaluation	7	7	-	3	17
OM debugging and repair	5	6	-	3	14
OM comp. corr. and pat.	3	1	-	1	5
OM corr. representation	1	-	-	1	2
OM application	1	2	-	2	5
Ontology similarity	1	-	-	-	1
Ontology learning	-	1	-	-	1
Ontology reasoning	-	-	-	1	1
Ontology modularization	-	-	1	-	1
Ontology evaluation	3	-	-	1	4

several papers using OntoFarm in other semantic web subfields: *ontology similarity*, *learning*, *reasoning*, *modularization* and *evaluation*, see Table 3.5.

Although most of the papers belong to the semantic web community, based on the distribution of papers publication venues[11] we can conclude that OntoFarm is also used in a broader community of artificial intelligence, i.e., AAAI (Conference on Artificial Intelligence), ECAI (European Conference on Artificial Intelligence) and IJCAI (International Joint Conference on Artificial Intelligence) conferences and even in other communities according to papers within e.g., *Neurocomputing* journal and *Computes & Chemical Engineering* journal.

OntoFarm became an important ingredient in many EU and national projects. The

[11] Numbers of publication venues are available at `https://owl.vse.cz/ontofarm/venues.html`

benchmark has been employed for semantic web research within Knowledge Web[12] (realizing the semantic web; 2004-2007), TONES[13] (automated reasoning techniques for engineering; 2005-2007), BOEMIE[14] (evolving multimedia ontologies; 2006-2009), SEALS[15] (benchmarking semantic tools; 2009-2012), MONNET[16] (Multilingual Ontologies for Networked Knowledge; 2010-2013) and Optique[17] (semantic technologies for big data; 2012-2015).

Further, there have been two international projects that used OntoFarm for their evaluation part: eSYMBIOSIS[18] (development of knowledge-based web services; 2010-2013; Great Britain, Greece) and CAMELEON[19] (multilingual lexica and ontologies; 2011-2014; Brazil and France). At national level, OntoFarm has been used in different projects within 11 countries.

3.5 Practitioner-Oriented Survey

In order to find out what is the current position and future prospects of OntoFarm according to its users, I performed a questionnaire survey between November 2015 and January 2016.

3.5.1 Participants and Survey Design

The participants of the survey have been OntoFarm users. I identified these users during my literature review. In all, I collected 130 unique contact emails. The questionnaire was prepared as a Google form, and the link to it was distributed to all 130 contacts by email. In 27 cases I received an automatic reply that the contact email does not exist any more. In all I received 12 answers.

[12] `http://knowledgeweb.semanticweb.org/`

[13] `http://www.inf.unibz.it/tones/`

[14] `http://cordis.europa.eu/ist/kct/boemie_synopsis.htm`

[15] `http://www.seals-project`

[16] `http://cordis.europa.eu/project/rcn/93713_en.html`

[17] `http://optique-project.eu`

[18] `http://www.esymbiosis.gr`

[19] `http://cameleon.imag.fr`

Table 3.6: Each question, its answer options and survey results are listed in several rows. The question itself is written in the first row (marked with the order number). Answer options (here shortened) are listed in the next rows. Results from the survey are stated in braces, for each option. Answer options with highest results per question are underlined. T. stands for the type of the question, where 1 means that the participant could only select one answer option, while M means that any number of answer options could be selected. In case the respondent provided a negative answer for the question (no. 3 and 8), (s)he was asked to specify why (this is depicted with the '+' symbol). Source: [84].

No.	Question, answer options and results	T.
1.	How did you learn about the OntoFarm collection? {*personal recommendation (17%), OntoFarm paper 2005 (0%), another research paper (0%), OAEI (83%), MultiFarm (0%), don't remember (0%)*}	1
2.	What have been the features for which you or your team chose OntoFarm for your project? {*suitable domain (42%), relatively expressive ontologies (50%), presence of style heterogeneity (50%), availability of reference alignment (83%), no reason (0%), other (8%)*}	M
3.	Do you think that conference organization is a suitable domain for a widely usable experimental ontology collection? {*strongly agree (17%), agree (75%), undecided (8%), disagree (0%), strongly disagree (0%)*}	1+
4.	For which semantic web subfield have you used OntoFarm? {*matching (92%), learning (0%), development (8%), debugging (17%), evaluation (8%), reasoning (8%), modularization (0%), visualization (0%), other (8%)*}	M
5.	Have you used all ontologies from OntoFarm or only some of them? {*all of them (17%), only those included in a reference alignment (75%), other subset of OntoFarm (8%)*}	1
6.	What do you think OntoFarm needs in order to become a better experimental collection for you or your team? {*reference alignment extension (58%), improving naming of entities (25%), fixing trivial modeling issues (25%), adding more ontologies (33%), more informative web page (8%), other (25%)*}	M
7.	In case you think that ontologies in OntoFarm should be repaired, what procedure would you prefer? {*replacement of current ontologies by new ones (25%), new ontologies as an alternative variant (58%), no preference (8%)*}	1
8.	Do you plan to still use OntoFarm in the future? {*certainly (42%), probably (42%), undecided (17%), probably not (0%), certainly not (0%)*}	1+
9.	What are the most neglected ontology features that current ontology collections typically lack in general? {*ontologies with high expressivity (33%), large ontologies (50%), multilinguality in ontologies (17%),* {*ontologies having rich annotations in natural language (25%), other (17%)*}	M

The survey was designed with 9 questions, all being multiple-choice ones. Six participants out of 12 deliberately disclosed their identity by providing their email for potential contact in the future. The questions from the survey (including their answer options and results from the survey) are presented in Table 3.6. The survey attempted to find out about typical dissemination channels (Question 1), typical motivation to use OntoFarm (Question 2), the degree of agreement regarding suitability of the selected domain (Question 3), the typical application field (Question 4), OntoFarm ontologies that are most often used (Question 5), typical shortcomings of OntoFarm and degree of agreement on its potential improvement strategy (Questions 6 and 7), whether OntoFarm users will further use it in future (Question 8) and ontology features that ontology collections typically lack in general (Question 9).

3.5.2 Discussion of Survey Results

This section presents summarized conclusions based on the survey results, while the detailed results are available on a web page.[20]

Answers to Question 1 indicate that 83% (10 of 12) respondents learned about OntoFarm from OAEI. This result shows that the OntoFarm collection is mostly popular thanks to OAEI, where it has been included from its inception, within the conference track. This is also the reason why the collection is often better known as 'the OAEI conference dataset'.

The results of the survey also confirm the proper initial motivation of creating the OntoFarm ontology collection. The answers to Questions 2 and 3 verified the relevance of the collection's intrinsic characteristics. The importance of the characteristic *sharing the same, generally understandable domain* is indirectly supported by the answers to Question 3, where 92% (11 of 12) participants (strongly or less strongly) agreed that conference organization is a suitable domain. The importance of the characteristic *being built by different groups, thus naturally mimic different conceptualizations* has been verified by Question 2, where the presence of modeling style heterogeneity was a reason to use OntoFarm for half of the re-

[20] `https://owl.vse.cz/ontofarm/survey-results/`

spondents. Finally, the importance of the characteristic *being rich in various types of axioms* has been, again, verified by Question 2, where the feature of 'containing relatively expressive ontologies' was a reason to use OntoFarm for half of the respondents.

The above mentioned intrinsic characteristics of OntoFarm had been achieved thanks to the effort of providing a high-quality dataset for ontology matching. The results of Question 4 of the survey confirmed that this would be the most common usage of OntoFarm since 92% (11 of 12) respondents used OntoFarm in the ontology matching field. On the other side, we could also find other semantic web subfields where OntoFarm was involved. This corresponds to my literature analysis to a certain extent, and is also obviously interconnected since the respondents were authors of the analyzed papers dealing with OntoFarm.

Although OntoFarm did not have a reference alignment for its first two years, we can clearly see that the creation of a reference alignment throughout years 2008-2010 (and in 2015) was a very important benefit for the community. This was confirmed by the fact that 83% (10 of 12) respondents (Question 1) stated that the availability of a reference alignment was the reason to use OntoFarm. This is an obvious effect since a reference alignment enables people to evaluate the performance of their matching systems. Significance of reference alignment was further confirmed by answers to question 5 where 75% (9 of 12) respondents answered that they use only those ontologies which are involved in reference alignments.

The importance of reference alignment is further amplified by the answers to Question 6, where 58% (7 of 12) respondents considered that by extending the reference alignment OntoFarm could become a better ontology collection. Its 16 ontologies seem to be a kind-of optimal size since only 4 respondents call for adding more ontologies from the same domain. One respondent had an interesting idea of 'adding one large ontology only partially overlapping and creating alignments to it'. This could establish a new test case. Current test cases within the conference track of OAEI contain similarly large ontologies, which overlap for almost 100%. The matching systems thus can apply a relatively simple strategy to match almost all entities from one ontology to another one. A large and partly overlapping ontology could help evaluate matching systems with regard to their capability to only match similar parts of an ontology.

Although the wish of having the ontologies repaired was not so strong according to Question 6 – 25% (3 of 12) for naming repair and 25% for repair of trivial modeling issues – there has been relative consensus on the repair procedure. 58% (7 of 12) respondents agreed that the repaired ontologies should be made available as a variant of the original collection. It would make sense to keep both the original variant and the repaired one since mistakes are natural in real ontologies as well. One respondent also suggested an interesting idea of "documenting the modeling issues, since modeling errors are common and thus it is good to have some reflected in the ontology collection, but it would be helpful to know which ones are known and to be expected".

The utility of OntoFarm for the community can be also inferred from the fact that 83% (8 of 12) respondents plan to use OntoFarm in the future, where half of them is certain about this and half of them thinks it is probable.

Finally, answers to Question 9 show that OnfoFarm users (half of them) think that the lack of large ontologies is the most important gap in current ontology collections, in general. Further, 33% (4 of 12) respondents call for collections with more expressive ontologies and 25% (3 of 12) for collections with ontologies equipped with rich annotations in natural language. Two respondents added the desire for a collection including both a significant TBox and ABox.

3.6 Chapter Summary

The survey results confirmed that the initial requirements on OntoFarm had been well chosen and that OntoFarm fills well its role of experimental ontology collection. Last but not least, I collected the suggestions raised from the survey which could lead us to further improve the usability of the collection.

Based on literature analysis, on the one hand the OntoFarm benchmark became an important benchmark in many projects and for many researchers. On the other hand the updated literature analysis (since the paper [84] has been published) showed that there is a significant descrease of interest in OntoFarm.[21] This can be explained by the fact that the most recent research deals with instance matching, large-scale matching and complex matching. While former two are out of the scope of OntoFarm, the latter has been prepared based on OntoFarm as its derivative.

[21] I could not analyze some papers in deep because I had not had the access to them. However, based on their abstracts I assume that those papers did not deal with OntoFarm.

Chapter 4

OOSP: Support of Ontology Tool Benchmark Construction

An automatic ontology tool benchmark construction process can be supported by my approach, described in this chapter, implemented in the “Online Ontology Set Picker” (OOSP) tool available at `https://owl.vse.cz/OOSP/`. Thus, people involved in the benchmarking of ontology tools (either the benchmarking organizers such as people from the OWL Reasoner Evaluation workshop[1] or directly the creators of ontology tools) are potential users of OOSP. This chapter provides the details about the OOSP system as a contribution of the author and the main practical outcome of the COSOL project (succesfully evaluated the CSF post-doctoral project, 2014 - 2016), by presenting the popular ontology resources related to this work (Section 4.1), the tool’s overall architecture allowing for ontology storing and indexing ontologies according to ontology metrics metadata (Section 4.2.1), the search approaches, the corresponding OOSP components and workflows in OOSP with regard to its HTML interface (Section 4.2.2), a summary of the interviews with experts (Section 4.4.1), a report about an experiment performed by users (Section 4.4.2), and finally, a summary of related work (Section 4.5) and the conclusions of the chapter together with the topics for

[1] `https://www.w3.org/community/owled/workshop-2016/`

future work (Section 4.6).

OOSP has been described by Zamazal and Svátek in three successive demo papers: OOSP itself was introduced in a demo paper at the SumPre 2015 workshop in [80], the categorization-power-based search was introduced at the SumPre 2016 workshop in [83], and the similarity-based search approach was introduced at the SEMANTiCS conference during the demo and poster session in [82]. This chapter provides an overall description of the approach, including new parts such as the lexical-token-based search and the report about an experiment performed by users who followed test scenarios focusing on the tool's usability, usefulness and comprehensibility. Most of the content of this chapter has been submitted to the International Journal of Metadata, Semantics and Ontologies (indexed by Scopus) where it is currently in the second round of the reviews.[2]

4.1 Ontology Resources

Ontologies can be found in ontology resources – either within ontology repositories or via ontology search engines retrieving ontologies on the web. Historically, the first ontology search engine was *Swoogle* presented by [16]. The Swoogle Semantic Web search engine extracts metadata for each document and computes the relations among them. Nowadays Swoogle indexes almost 4 million semantic documents and allows users to search for ontologies and their instances within this index. Swoogle provides the search using keywords applicable on classes and properties where Boolean operators (AND, OR and NOT) can be used. Further, by using RDF metadata one can restrict numbers of triples, classes, properties or instances.

One of the most famous ontology search engines is *Watson* introduced by [15]. Watson crawls ontologies and other semantic documents from the web and enables the ontology

[2] According to the permissions described on the web page of the Inderscience publisher, authors can use their article for non-commercial purposes after publication in other works by the Author as stated at `http://www.inderscience.com/info/inauthors/author_copyright.php`. If the paper is accepted, Inderscience will be contacted for requesting of the explicit reproduction permission.

search with keywords from different ontology aspects, e.g., labels. Via its Java API, Watson also provides a SPARQL endpoint along with some pre-computed metrics metadata: concept coverage, DL expressivity, representation language (e.g., RDFS), numbers of classes, properties, individuals, and statements. However, ontologies are not searchable according to those ontology metadata. Finally, Cheng et al. presented *Falcons*, the semantic web search engine in [14], which provides a keyword-based search for objects, concepts (classes and properties), ontologies, and RDF documents on the web. Except for Swoogle, which provides ontology searching only using the basic ontology metrics metadata, such as the numbers of triples, classes, properties and instances, these ontology search engines do not provide searching based on a rich set of ontology metadata.

There are several prominent ontology repositories collecting high-quality ontologies. *BioPortal* introduced by [77] is a library of well-curated biomedical ontologies. The current release[3] contains 726 ontologies in different formats, including some adapted from another repository, the OBO foundry. BioPortal provides a term-based search for classes and properties in ontologies, where one can further restrict the ontology category (e.g., anatomy). BioPortal RESTful services offer several count-based metrics per ontology, e.g., the number of classes or properties. The BioPortal technology has recently been reused for agronomy domain ontologies to build *AgroPortal* presented by [46].

Another ontology repository, *Linked Open Vocabularies* (LOV) introduced by [73], collects ontologies rather according to their usage. LOV is a well-curated collection of linked open vocabularies used in the Linked Data Cloud. Currently there are 650 ontologies covering diverse domains, e.g., publications, science, business or city. The ontologies/vocabularies are usually small and they are used within diverse linked open data applications. LOV also provides a RESTful service for a term-based search over ontologies or terms, and a SPARQL endpoint.

Although some repositories present the relevant values of ontology metrics metadata,

[3] By "currently" I always mean on August 27, 2018 throughout the chapter.

these are not searchable. Other ontology repositories solely provide collections of ontologies without rich metrics metadata (e.g., the Oxford Ontology Library,[4] Protégé Ontology Library[5] or Ontohub[6]). In all, there has not been a way to search ontologies from existing repositories based on different ontology metrics metadata. OOSP fills this gap.

4.2 The Architecture of OOSP

The OOSP system is divided into two parts as depicted in Figure 4.1. Its back-end deals with ontology gathering from ontology repositories and computation of ontology metrics metadata, see Section 4.2.1, while the front-end enables the user to construct an ontology benchmark corpus by implementing different search approaches, see Section 4.2.2. OOSP is a web-based application implemented using Java Servlet Pages, JavaScript and OWL-API.[7] Ontologies with their imports are stored on disk and ontology metrics metadata values and statistics are stored in a MySQL database.

4.2.1 The Back-end of OOSP

The OOSP back-end consists of three components, all relying on the OWL-API: *Downloader*, *OntologyProcessor* and *StatsCounter*. The Downloader is responsible for ontology gathering from their original repositories into particular ontology pools. Each ontology is stored under a unique identifier (storage code) in a file on a hard drive and its record is created in the relational database ONTOLOGIES table. This component also downloads the imports closure of the given ontologies. The ontology snapshots in OOSP are thus available even if the imported ontologies are not accessible any more. For enabling this there is created mapping table relating the original URI of imported ontology to its unique storage code. The benefits

[4] `http://www.cs.ox.ac.uk/isg/ontologies/`

[5] `http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library`

[6] `https://ontohub.org/`

[7] `http://owlapi.sourceforge.net/`

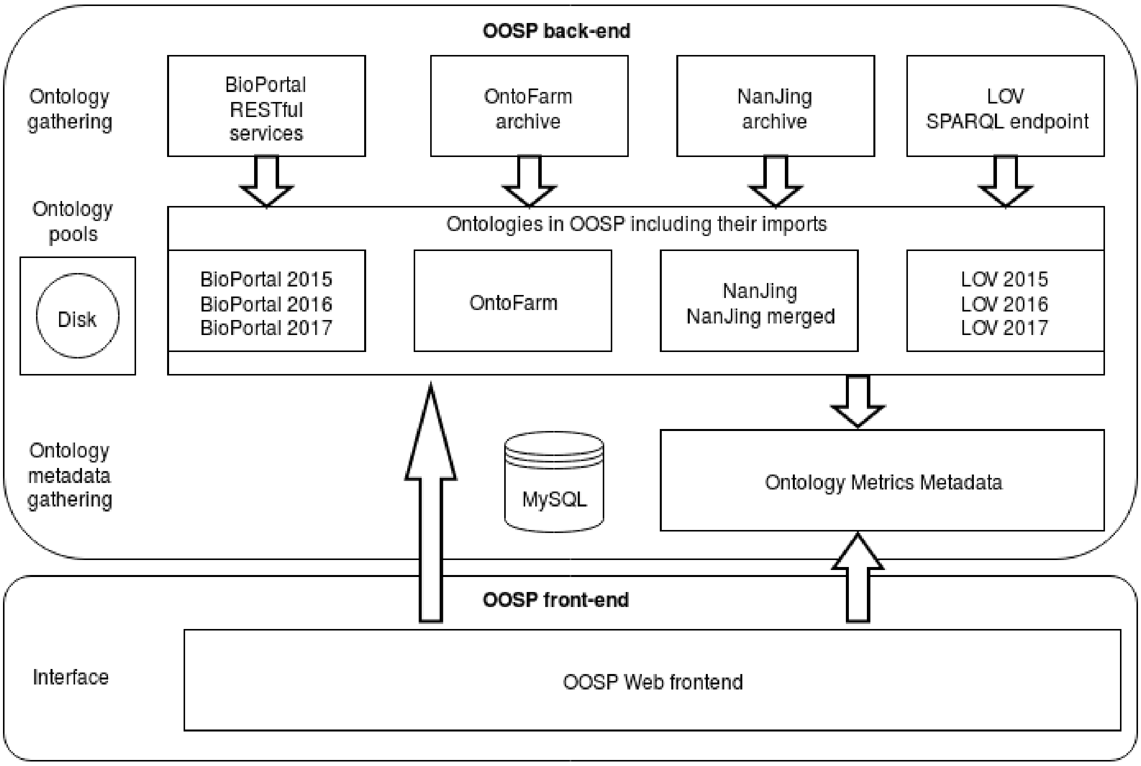
OOSP back-end
Ontology gathering
BioPortal RESTful services
OntoFarm archive
NanJing archive
LOV SPARQL endpoint
Ontology pools
Ontologies in OOSP including their imports
Disk
BioPortal 2015 BioPortal 2016 BioPortal 2017
OntoFarm
NanJing NanJing merged
LOV 2015 LOV 2016 LOV 2017
Ontology metadata gathering
MySQL
Ontology Metrics Metadata
OOSP front-end
Interface
OOSP Web frontend

Figure 4.1: The OOSP architecture. Source: Author.

from the imports availability are for both the OOSP system where additional metadata metrics can be added later on and for the users who can get ontology with its imports. This component varies according to what kind of access the given repository supports. Basically, there are three different access policies implemented: *REST-based* access, *SPARQL-based* access and *file-based* access. REST-based access enables easy access to ontologies, e.g., in BioPortal. SPARQL-based access means downloading the ontologies using a SPARQL front-end, e.g., in LOV. Several ontology repositories make their ontologies available via zip archives, e.g., the OntoFarm and Nanjing repositories.

The OntologyProcessor component processes downloaded ontologies and their downloaded imports in order to gather ontology metrics metadata. Metrics are added to the corresponding ontology records in the ONTOLOGIES table. The ontologies in OOSP are *indexed* by the computed values of ontology metrics and by tokens from the local names, labels and comments of entities. The ontology metrics metadata are divided into eight groups covering the most important ontology aspects. In all, there are 104 different kinds of ontology metrics (I state the number of different kinds of metrics in parentheses). *Entity* metrics (9) include numbers of entities (e.g., classes, instances); *axiom* metrics (27) include numbers of different axiom types (e.g., subsumption, equivalence); *class expression type* metrics (11) include expression types used for the construction of anonymous classes (e.g., existential quantification); *taxonomy* metrics (9) include the characteristics of the taxonomy (e.g., the number of top classes, leaf classes or the maximum taxonomy depth); *OWL2 profiles and reasoning* metrics (7) include the profile information along with information about the consistency and the number of unsatisfiable classes[8] *annotation* metrics (6) include the counts of selected annotation types (e.g., labels or comments) and of different languages involved in label annotations; *detail* metrics (13) include some newly designed metrics metadata related to domain/range (e.g., the number of anonymous classes as a domain definition); finally, there are *categorization-power-based* metrics comprising the absolute and relative numbers

[8] I applied the HermiT reasoner: `http://hermit-reasoner.com/`

of different categorization options an ontology provides (11) and the absolute and relative numbers of focus classes according to five categorization options (11). Categorization options and focus classes are related to Approach 4 explained in Section 4.2.2.

The StatsCounter component computes the basic descriptive statistics for each ontology pool: the ratio of ontologies having at least one occurrence of the object aggregated by the metric (for binary metrics such as OWL profiles, it is simply the ratio of positive values), the ratio of ontologies for which respective metrics is unknown (*N/A*), e.g., the reasoner could not process some ontologies due to unsupported datatypes; and the descriptive statistics (median, average, standard deviation and maximum) of the metrics over all ontologies.[9]

Ontology Repositories Available in OOSP

Currently, there are four ontology repositories available in OOSP in nine snapshots called "ontology pools":

- Three *BioPortal* pools: the *BioPortal 2015* pool contains 317 ontologies (85%) out of 420 ontologies from the BioPortal February 2015 snapshot, the *BioPortal 2016* pool contains 399 ontologies (80%) out of 501 ontologies from the BioPortal January 2016 snapshot and the *BioPortal 2017* pool contains 494 ontologies (75%) out of 657 ontologies from the BioPortal November 2017 snapshot. BioPortal contains ontologies in different formats, including some adapted from another repository, the OBO foundry.[10] Some ontologies were not successfully processed due to different reasons such as the 'not found' error, private access, unavailable imports or parsing problems using OWL-API (a format problem or the impact of very large ontologies).

- Three *LOV* pools: the *LOV 2015* pool contains 461 (97%) out of 475 ontologies from the LOV February 2015 snapshot, the *LOV 2016* pool contains 509 (96%) out of 529 ontologies from the LOV January 2016 snapshot, the *LOV 2017* pool contains

[9] The minimum is omitted since it is usually zero.

[10] `http://obofoundry.org/`

568 (92%) out of 617 ontologies from the LOV November 2017 snapshot. The most common problems were due to parsing by OWL-API and unavailability of imports.

- The *Nanjing* pool contains 1403 ontologies extracted from single files and 225 ontologies extracted from more than one RDF file. This corresponds to the experimental ontology pool *Nanjing merged* which contains ontologies created by merging their definitions spreading over RDF files. Out of 1763 ontologies (Jan. 2016 snapshot), 135 were not parsed by OWL-API or were not processed due to unavailable imports.

- Finally, I provide the *OntoFarm* pool based on the *OntoFarm* ontology collection, which includes 16 small but relatively rich ontologies from the conference organization domain. The collection has previously been used for experiments in Ontology Matching and elsewhere [84] and is described in Chapter 3.[11]

In order to characterize the nine ontology pools available via OOSP, I display the means of selected characteristics from the ontology metrics metadata in Table 4.1. Since the ontology pools usually contain outliers distorting the overall statistics of the pool, I first detect the outliers using *Tukey's method* (explained by [72]) with its *interquartile range* approach. This method was used for each ontology metrics metadata where the detected outliers were replaced with "NA" values.

According to Table 4.1 the BioPortal ontologies (from 2015, 2016, 2017) typically have many more classes (798, 688, 618) organized in more layers (8, 7, 8) with many top classes (21, 14, 11), leaf classes (663, 530, 470) and subclasses (963, 795, 827) than ontologies from the other ontology pools. The LOV and NanJing ontology pools are very similar to each other in that they contain rather small ontologies (on average 20 resp. 12 classes) featuring no anonymous classes in the domain and range. Finally, OntoFarm, except for the class-related characteristics, dominates all other presented ontology metrics metadata with regard to their means.

[11] `https://owl.vse.cz/ontofarm/`

Table 4.1: Means for selected statistics for nine ontology pools. The highest values are in bold. Source: Author.

metric	BioPortal 2015	BioPortal 2016	BioPortal 2017	LOV 2015	LOV 2016	LOV 2017	NanJing	NanJing Merged	OntoFarm
classes	**798**	688	618	19	20	20	12	6	44
object properties	12	12	13	16	16	15	6	2	**33**
datatype properties	1	1	1	6	6	6	2	1	**12**
instances	0	2	2	4	4	3	2	**8**	2
layers	**8**	7	**8**	3	3	2	2	1	4
top classes	**21**	14	11	8	9	9	5	3	7
leaf classes	**663**	530	470	16	16	16	10	5	33
subsumptions	**963**	795	827	13	13	13	7	1	67
multiple inheritance	0	0	**1**	0	0	0	0	0	0
named domain	1	2	2	13	15	14	4	0	**32**
anonymous domain	0	0	0	0	0	0	0	0	**3**
named range	1	1	2	10	10	10	3	1	**27**
anonymous range	0	0	0	0	0	0	0	0	**2**

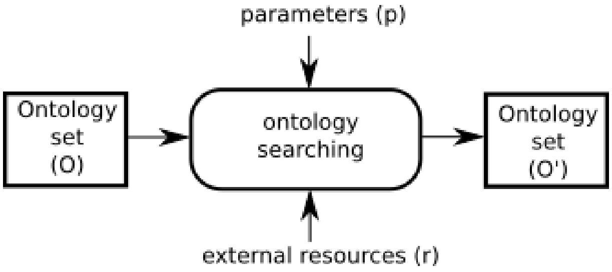

Figure 4.2: The ontology search. Source: Author.

Table 4.1 also shows that the BioPortal snapshot 2015 contains larger ontologies (not considering outliers) than the BioPortal snapshots from two following years, which means that in 2015 the process was more successful to manage large ontologies. Further, we can see that while for the LOV and NanJing snapshots the average number of subsumptions is lower than the average number of classes per ontology, this does not hold for the BioPortal and OntoFarm snapshots. This can be explained by the fact that some ontologies can have classes stating their subsumption to the most general owl:Thing even if those classes are in lower layer of the class taxonomy or/and some ontologies can have classes with partial definition(s) using subsumption(s) to anonymous class(es). We can also observe high numbers of leaf classes, which can be explained by the fact that on average the ontologies are usually rather flat.

4.2.2 The Front-end of OOSP

In order to create an ontology benchmark, OOSP applies an ontology search process. Technically, the *process of ontology searching* can be defined as *a function f which, from a set of ontologies O to search in, a set of parameters p and a set of external resources r, returns a set of ontologies O'*:

$$O' = f(O, p, r) \tag{4.1}$$

This can be schematically represented as illustrated in Figure 4.2. OOSP supports four distinct ontology search approaches (referred as Approaches), which differ in their types of input parameters. An external resource is used only in a similarity-based approach (Approach 2 below). For all approaches, the set of ontologies on input corresponds to the selected ontology pool. The ontology set on the output is recognized as a potential *benchmarking corpus.*[12] These approaches are accessible via the OOSP front-end. In order to increase tool usability, there are also available screencasts. In the OOSP front-end, there are basically two general components. The *Restriction Selection Component* covers the following functions:

- setting the ontology pool,
- setting specific types of restrictions.

Further, the set of ontologies on output is realized by the *Benchmarking Corpus Component* having the following functions:

- getting a benchmarking corpus meeting the chosen restrictions along with their ontology metrics metadata,
- downloading each ontology from the corpus, ontology with its imported ontologies as a zip archive or ontology merged with its import closure,
- downloading the whole ontology metadata metrics table and the summary descriptive statistics,
- getting visualizations of distributions of ontology metadata metrics.

Each ontology search approach extends these two general components and they are used in different workflows via HTML interface as described in the following sections, namely in "Component and Workflow" paragraphs.

[12] Throughout this chapter I use notions "benchmarking corpus" and "ontology benchmark" interchangeably. However, if we look closer, the benchmarking corpus is the output of OOSP and it is expected that it is further adjusted before being fully-fledged ontology benchmark.

Approach 1: Based on Exact Intervals of Metrics

In principle, in this case of *Ontology Search based on Given Exact Intervals of Metrics*, the user can choose as parameters any ontology metrics and specify the minimum, maximum or equality restriction. These restrictions can be combined, which has the semantics of conjunction. On the output there is a *benchmarking corpus* meeting the restrictions.

Components and Workflow This approach uses a general benchmarking corpus component. Extension of the restriction selection component, namely the *Metrics Selection Component*, contains of the specific function: setting restrictions as exact intervals of ontology metadata metrics.

It is available for end-users via the HTML interface in a three-step workflow depicted in Figure 4.3 a). First, the initial *ontology pool* is selected. Second, the user can browse through the eight ontology metadata *metric* types, see Section 4.2.1, and specify values (*max* and/or *min*, except nominal values such as OWL profiles) for individual metrics. To make the restriction setting more informed, the following statistics are provided: the ratio of ontologies having at least one occurrence of the object aggregated (via count, average or max) by the metric; the ratio of ontologies for which respective metrics is unknown (*N/A*); and descriptive statistics (median, average, standard deviation and maximum) of the metric over all ontologies. Third, the user obtains the *benchmarking corpus* meeting the provided restrictions. For the resulting benchmarking corpus OOSP provides a table containing all metrics values for all selected ontologies. An ontology from the set can be downloaded in three ways: one *separate* ontology as an OWL file, one ontology with *all ontologies from its import closure* as a ZIP archive, or an *ontology merged with its import closure* as one OWL file. There are further three ontology-set-wise download options: only the table (in CSV); ontology set summary descriptive statistics (also in CSV); and actual ontologies as OWL files (ZIP archive). Finally, for selected eight metrics (classes/instances counts, axiom types, DL constructs, OWL 2 profiles, annotations, domain/range definition types) OOSP also offers

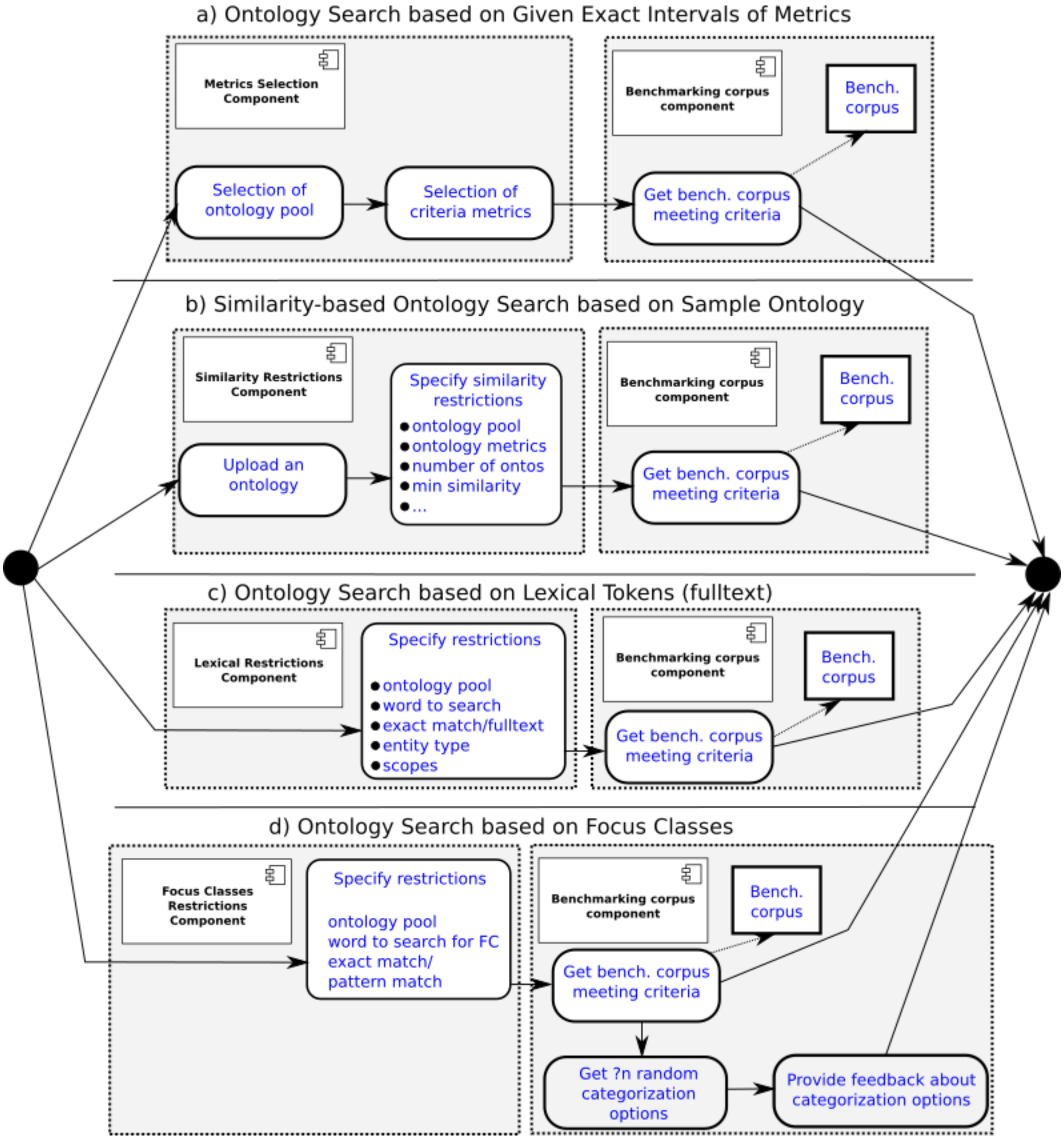
a) Ontology Search based on Given Exact Intervals of Metrics
Metrics Selection Component
Selection of ontology pool
Selection of criteria metrics
Benchmarking corpus component
Bench. corpus
Get bench. corpus meeting criteria
b) Similarity-based Ontology Search based on Sample Ontology
Similarity Restrictions Component
Upload an ontology
Specify similarity restrictions
●ontology pool
●ontology metrics
●number of ontos
●min similarity
●...
Benchmarking corpus component
Bench. corpus
Get bench. corpus meeting criteria
c) Ontology Search based on Lexical Tokens (fulltext)
Lexical Restrictions Component
Specify restrictions
●ontology pool
●word to search
●exact match/fulltext
●entity type
●scopes
Benchmarking corpus component
Bench. corpus
Get bench. corpus meeting criteria
d) Ontology Search based on Focus Classes
Focus Classes Restrictions Component
Specify restrictions
ontology pool
word to search for FC
exact match/
pattern match
Benchmarking corpus component
Bench. corpus
Get bench. corpus meeting criteria
Get ?n random categorization options
Provide feedback about categorization options

Figure 4.3: The components and workflows of the OOSP front-end. Bench. stands for benchmarking. Source: Author.

graphs of ontology set statistics. Further, a user can ask for a randomly selected subset of benchmarking corpus with required cardinality (this option is currently not available for other ontology searches).

Approach 2: Based on Ontology Similarity

Similarity-based Ontology Search uses a sample ontology as an external resource in the searching process. Further, it is possible to select as parameters any combination of ontology metrics metadata, the maximum number of ontologies on output, the minimum and/or maximum of the similarity threshold, and the preference of the upper or lower end of the similarity interval.

In the case of Approach 2 (introduced by [83]) OOSP, for similarity computation, employs the R language.[13] First, OOSP computes the ontology metrics metadata for a sample ontology and it generates an $n \times m$ data matrix for the given ontology and the selected ontology pool, where n corresponds to the number of ontologies in the selected ontology pool plus one seed ontology given by the user and m to the number of the selected ontology metrics. Regarding the similarity computation, OOSP first applies the *scale* function in R, corresponding to the normalization by the *standard score* (z-score), on the input data matrix. Next, it computes the *Euclidean distance* between the given ontology and all ontologies from the selected ontology pool in a pair-wise manner, where the two vectors v_1, v_2 represent ontologies O_1 and O_2 with values of the computed ontology metrics metadata:

$$d(v_1, v_2) = \sqrt{\sum_{j=1}^{m} (v_{1j} - v_{2j})^2} \quad | \quad d(v_1, v_2) \in [0, \infty]. \tag{4.2}$$

Distance is then transformed to similarity using the following formula:

$$s(v_1, v_2) = \frac{1}{1 + d(v_1, v_2)} \quad | \quad s(v_1, v_2) \in [0, 1]. \tag{4.3}$$

[13] `https://www.r-project.org/`

Finally, the ontologies for the ontology set on output are selected according to the similarity search parameters, i.e., the maximum number of similar ontologies, the similarity interval, and its preferred end.

Components and Workflow While the *Benchmarking Corpus Component* is in its basic shape, the restriction selection component, namely the *Similarity Restrictions Component*, is equipped with the following specific functions:

- uploading/selecting seed ontology,
- setting dimensions of space for similarity computing by selecting ontology metrics metadata,
- setting specific similarity restrictions as stated above.

It is available for end-users via the HTML interface in a three-step workflow depicted in Figure 4.3 b). First, the user provides either the *URI of an online ontology* or a *storage code* of an ontology stored in OOSP. Second, s/he obtains an *ontology metrics overview* for the given ontology; while for an online ontology they are computed on the fly, for ontologies stored in OOSP they are merely retrieved. S/he may select an *ontology pool* restricting the scope of ontologies to be considered. Next, s/he selects the ontology metrics to be considered for similarity computation out of six ontology metrics groups: entity metrics, axiom metrics, class expression type metrics, taxonomy metrics, annotation metrics and finally, detail metrics (see Section 4.2.1). As further parameters s/he can specify the maximum number of ontologies on output, the minimum and/or maximum of the similarity threshold, and the preference of the upper or lower end of this interval. Third, the user obtains the *benchmarking corpus* meeting the provided restrictions with the all options as described above for Approach 1.

Approach 3: Based on Lexical Tokens

Ontology Search based on Lexical Tokens requires an entity token as a parameter on input and a further specification about the exact match or word match (i.e., in full-text), in what kind of entities (classes, properties, individuals) and in which scope (local name, label or comment) the search should take place. Full-text only works in the all-scope mode. Further, if a user selects the "word match" option an ontology search is performed in a full-text natural language mode. This is supported by using MySQL, where a full-text index for the selected columns (local name, label and comment) is established. In all, currently there is an index comprising over 13 million entries searchable in seconds.

Components and Workflow The *Benchmarking Corpus Component* is extended with the possibility of getting the list of found entities for each ontology in the resulting benchmarking corpus. The extension of the restriction selection component, namely the *Lexical Restrictions Component*, is equipped with the function: setting lexical restrictions. It is available for end-users via the HTML interface in a two-step workflow depicted in Figure 4.3 c). First, an initial *ontology pool* is selected and a word for fulltext search is provided by the user; the word can be searched as a whole localname or as a part of it and further restrictions can be selected as described above. Second, the user obtains the *benchmarking corpus* meeting the provided restrictions with the all options as described above for Approach 1.

Approach 4: Based on Focus Classes

Ontology Search based on Focus Classes is an analogy to Approach 3; however, in this case, lexical tokens are restricted on focus classes (FCs) and, besides the entity token on input, there is only a specification about the exact or pattern match (i.e., %word%) to be used. This approach was introduced in demo paper by [82]. The key concept for this approach is a *focus class* which is a class for which the ontology provides categorization options (COs). The most typical categorization option is captured as subclasses of a focus class, e.g., *Men*

SubClassOf: Person. However, there are other categorization options, which can be written in the form of compound concept expressions, e.g., *(bornIn value Italy)* $\sqsubseteq$ *Person*, hinted by the ontology structure. Categorization options of focus classes are more rigorously explained by [68]. The assumption is that in such ontologies the given concept plays a more central role. In order to enable this approach, the first categorization options of classes for all ontologies were generated. I processed all ontologies from all repositories included in OOSP except BioPortal because ontologies from BioPortal usually have a large size.

Components and Workflow The *Benchmarking Corpus Component* is extended with the following functions:

- getting two separate ontology sets; the first providing the ontologies containing FCs according to the input keyword, and the second providing ontologies with classes (non-FCs) matching the keyword but having no COs,
- getting the list of found FCs for each ontology.

The extension of the restriction selection component, namely the *Focus Classes Restrictions Component*, is equipped with the function: setting focus class restrictions. It is available for end-users via the HTML interface in a two-step or four-step workflow depicted in Figure 4.3 d). First, an initial *ontology pool* is selected and a keyword for FC search is provided by the user; the keyword can be searched as a whole localname or as a part of it. Second, the user obtains ontologies divided into two tables corresponding two ontology sets as described above. For both tables OOSP also provides ontology metrics values. Showing not only FCs but also non-FCs can be of interest especially in single-domain collections. Further, it is possible to show relevant COs. Since there can be a huge number of COs, the user can ask for a random sample of size between 10 and 100 COs. The fourth, optional step consists of providing feedback to the system on which COs are not proper ontology categorizations.

4.3 OOSP Solutions for Benchmarking Construction Scenarios

Four scenarios have been described in Section 1.2. I will discuss these scenarios with regard to search approaches available in OOSP.

Scenario 1 has been described as benchmarking of language interoperability and coverage of various language combinations in ontology tools. This scenario can be partly solved using Approach 1 where axiom type and class expression type ontology metadata metrics are suitable for retrieving a benchmarking corpus aiming at language interoperability. However, due to the fact that we cannot specify language combinations in Approach 1 we cannot solve the request for coverage of various language combinations.

Scenario 2 has been described as benchmarking of ontology alignment tools requiring pairs of ontologies from the same domain and a reference alignment between them. This scenario can be supported using Approach 1 and 4 (this is available from "Ontology Search based on Given Exact Intervals of Metrics" using "Metrics Based on Focus Classes" from "Entity metrics") where we can set a certain keyword for focus class search, e.g., "book" and further exact intervals of ontology metrics, e.g., ones dealing with multilinguality or taxonomy metrics. Since found ontologies would have focus class named "book", it can be assumed that all of them are somehow about a book domain. A discovery of the reference alignment is not covered by the Approach.

Scenario 3 has been described as benchmarking of ontology reasoning tools which typically requires an ontology having non-trivial concept expressions. This scenario is supported with Approach 1 where non-trivial concept expressions can be reflected specifying axiom and class expression ontology metrics as well as OWL 2 profiles.

Scenario 4 has been described as benchmarking of ontology visualization tools which typically need ontologies covering all language constructs of the OWL language. This can be again supported with Approach 1 using entity, axiom, class expression and annotation ontology metadata metrics. Benchmarking of ontology visualization tools could also take advantage of benchmarking ontologies varying with regard to taxonomy metrics.

Although I only provide abstract solutions for each scenario, they already represent usage examples of OOSP. Further examples are provided in Section 4.4.2 where several test scenarios were described for testing of OOSP by a group of users.

4.4 Feedback from Expert and Non-Expert Users

In order to evaluate OOSP I targeted two groups: experts on benchmarking of three different ontology tool categories and non-expert users. While the evaluation with experts can mainly disclose usefulness and comprehensibility of the OOSP approach based on their expert testing of OOSP and their provided feedback via the interview, the experiment with non-expert users can rather provide usability insights based on their simultaneous interaction with OOSP fulfilling prepared testing scenarios and their provided feedback via the questionnaire. The evaluation with experts and the experiment with users is documented at `https://owl.vse.cz/OOSP-article/`.

4.4.1 Interview with Experts

In order to provide qualitative research, I contacted several experts in the field of ontology benchmark construction for different ontology tool categories and one expert in ontology tool comparison with regard to ontology visualization. Out of five ontology tool categories depicted in Section 1 I addressed three ontology tool categories: ontology authoring (interoperability), ontology reasoning and ontology visualization. I prepared 14 questions for an interview which covers the interviewee's background, as well as questions about search approaches and general features of OOSP and general remarks. The addressed experts had available information about the interview and their questions, `http://tinyurl.com/hdejaw2` and a brief introduction to OOSP, `http://tinyurl.com/he83mbw`. I conducted the interview with four experts: Nico Matentzoglu from University of Manchester on the 25^{th} of October 2016 as the expert for ontology reasoning benchmarking; Raúl Garcia Castro from Universidad Politécnica de Madrid on the 15^{th} of November 2016 as the expert for interoperability benchmarking; and two experts for ontology visualization where one wanted to remain anonymous and Dmitryi Pavlov from company VISmart Ltd, located in St. Petersburg, Russia.

I summarize their answers here, while their full answers are available online, `https://owl.vse.cz/OOSP-article/`. Nico has an extensive experience in constructing ontology benchmarks for reasoning, e.g., the ORE Reasoner competition 2015. He confirmed that there are some tools for generating benchmark ontologies, e.g., OntoBench, but there is no tool that enables experimenters to automatically generate benchmarking corpus sampled from collections of real ontologies. In this sense, OOSP goes in the right direction. He found OOSP important and useful as a sophisticated search engine for ontology corpus for generating benchmarks sets, but he also suggests that it would be better if 1) the underlying source corpus were complete, 2) unmodified (original serialization), and 3) they were up to date. While we can say that ontology pools available in OOSP are complete in a sense that missing ontologies from repositories were not possible to retrieve due to some technical issues (e.g., not parseable by OWL API), they are not unmodified (OOSP stores all ontologies in one serialization (RDF/XML)). Next, I do not provide up-to-date snapshots for BioPortal and LOV. This should be considered in future work. Further, the expert found search services and other services for working with found corpus in generally useful (except for the search based on focus classes, which did not convince the expert) and in addition he recommended integrating them all into one interface. In all, the OOSP web application is usable but it is a kind of old school design and a nicer web application would improve its use. The expert also confirmed that a wide range of metrics should be sufficient for most cases but he further recommended annotating ontologies with their proper names and domains. This could also be useful for searching. The expert also suggested some extension for similarity-based search, which is useful for getting similar ontologies, but it could be extended with more different similarity metrics (e.g., diff-based, signature overlap). Finally, from the expert perspective a random sub-sampling is sometimes useful, but in practice it is usually done by people constructing the benchmark and stratified random sub-sampling would also be good to add.

Raúl has an extensive experience with an ontology benchmark construction for different purposes; on the one hand efficiency and scalability and on the other hand conformance and interoperability. He found the tool nice, especially the search Approach 1. In his opinion, ontology metrics available in OOSP are useful but he also suggests to map them to a quality model to make them more usable for concrete scenarios. Although the search based on similarity did not convince the expert, he thinks that token based ontology search approaches (3 and 4) can be useful in general. But he thinks that selected ontology set would often need further preprocessing to be proper benchmark for a given scenario. Available OOSP services for working with ontology set are good (especially option to download the ontology set and quick analytics such as visualization and metrics) and he further suggests to add machine processable metrics and an alternative ontology selection approach when all ontologies could be selected by a user based on their assessement according to required values of metrics. The OOSP web application is basically usable but more sophisticated user interface would improve its use.

The anonymous expert has experience with an ontology benchmark construction for ontology visualization. He also confirmed that there are some tools for generating benchmark ontologies, e.g., OntoBench. He found all four search approaches useful and usable by stating that "OOSP has a good learning curve, not too steep" however the usability could be improved by better integration of search results with the rest of the user interface and by letting a user to navigate through all metrics in one view. He suggested to further refine ontology metrics, e.g., OWL 2 profile. In his opinion the provided services are useful but he asks for availability of Ontology Visualization Tool Recommender (OVTR) [79] in all ontology search options. OVTR is now only available in search Approach 1.

Dmitryi does not have experience with ontology benchmark construction but he is an expert on ontology visualization tools and their comparison; he participated in an ontology visualization survey in [21]. He was satisfied with all four search approaches as well as with the list of ontology metrics. He suggested to add some way to explore ontology in order to quickly see what the ontology really contains. This could be done in future using some external application for ontology visualization such as WebVOWL.[14] Regarding the usability of OOSP he suggested to redesign the user interface mainly with regard to metrics tables with optional number of visible metrics and sortable rows.

The expert interviews confirmed that OOSP is an important and useful tool; however, it can be improved to better achieve its goal, mainly in providing up-to-date snapshots and a nicer web application. The interviews provided an insightful feedback regarding extensions, which will be considered in our future work.

4.4.2 Users' Experiment

In order to complement the findings from the interviews with the experts with regard to further usability insights based on simultaneous interaction of more users with OOSP I performed users' experiments on the 27th of October 2016 to find out *usability*, *usefulness* and *comprehensibility* of OOSP and its four search approaches.

Experimental setting The participants of the experiment were Bachelor Degree students in a course on "Artificial Intelligence and Knowledge Representation". This course also provides an introduction to Semantic Web, OWL ontologies, ontology tools and their benchmarking. There were 13 students. Students were first instructed about theoretical background of OOSP and ontology benchmark construction. Next, the students were provided with a 30-minute overview of the practical usage of OOSP and its four ontology search approaches. Then they completed an assignment consisting of five *test scenarios* (TSs). The overall instructions as well as the tutorial for using OOSP were made available via

[14] `http://vowl.visualdataweb.org/webvowl.html`

a dedicated web-page, `https://owl.vse.cz/experimentOOSP/experiment-en.html`. TSs have been prepared so as to test difficulty level of comprehending the requirements and selecting the proper ontology search approach in OOSP. Full text of TSs is available at `https://owl.vse.cz/OOSP-article/testingScenarios.html`. Each TS asks the user to build a proper ontology benchmark:

- *TS1*: benchmark for testing ontology visualization tools on ontologies with large taxonomy with regard to a higher number of top and leaf classes and in which several natural languages are used in annotations.

  > *Exemplary Solution for TS1.* By using Approach 1 with top_classes>20 and leaf_classes>20 and multilinguality>1 and maxDepth>5.

- *TS2*: benchmark for testing an ontology populating tool which should have such ontologies from OntoFarm that enable categorization of the "Review" focus class.

  > *Exemplary Solution for TS2.* By using Approach 4 with specifying word "review", "exact" mode search within the OntoFarm ontology pool.

- *TS3*: benchmark for reasoners which need to be tested on ontologies having existential and universal quantifiers and which are consistent.

  > *Exemplary Solution for TS3.* By using Approach 1 with consistent=1 and constructSome>1 and constructAll>1.

- *TS4*: benchmark for testing an ontology learning tool using analysis on ontologies having annotations with "person" tokens.

  > *Exemplary Solution for TS4.* By using Approach 3 with specifying word "person", "fulltext" mode search with all entities and all scopes.

- *TS5*: benchmark for reasoners testing on 20 ontologies similar to the ekaw ontology with regard to numbers of class expressions and axioms.

> *Exemplary Solution for TS5.* By using Approach 2 with specifying the ontology 338585 (storage code) with all class expressions, all axioms types and maximally 20 similar ontologies.

Each user was asked to try to use the most suitable search option in OOSP on the required input ontology pool and store the resulted ontology benchmarks. The user also had to select proper restrictions based on a textual description; except TS1 where the user was provided with more detailed instructions about the ontology benchmark construction in order to initiate the user's work. In order to make the experiment fairer, the TSs are not in the order of numbering the search approaches.

In order to measure the time of accomplishment of each TS, the users had at their disposal a dedicated version of OOSP, which was augmented with five buttons in the main menu. After the user selects the button for the TS, the corresponding instructions to the test scenario are first displayed to the user, and OOSP logs the starting time. The time is also logged when the user retrieves the ontology benchmark and downloads it from OOSP. Students were asked to share their resulted ontology benchmarks for further inspection via the university information system. Students were also asked to report any issues with OOSP to the issue tracker of the OOSP project on GitHub. Finally, users had to answer several questions in the *questionnaire*.

The questionnaire was designed with six questions, out of which five were multiple-choice ones, and one was open. Additionally, each user had to provide the IP address of the computer used in the experiment. Thanks to this requirement, I can match the activity with OOSP and the answers in the questionnaire. The questions from the questionnaire (including their answer options and the results of the questionnaire) are presented in Table 4.2. The questionnaire attempted to find out about the general usefulness of OOSP (Question 1),

Table 4.2: Each question, its answer options, and the questionnaire results are listed in several rows. The question itself is written in the first row (marked with the ordinal number). The answer options are listed in the following rows. The results of the survey are stated in the parentheses for each option. The answer options with the highest results per question are underlined. A1 through A4 correspond to search approaches 1 through 4. WO (WO4) means the options for working with the output ontologies in TS1 to TS3 (in TS4 resp.). Question 6 was open and answers are indicated using "A:". Source: Author.

No.	Question, answer options, and results
1.	How useful is each search approach for ontology benchmark construction? (A1, A2, A3, A4)
	{*very useful (46%,23%,39%,23%), partly useful (54%,54%,39%,39%), slightly useful (0%,15%,23%,31%),*
	completely useless (0%,8%,0%,0%), did not understand (0%,0%,0%,8%)}
2.	How usable is each search approach and options for working with output ontologies? (A1, A2, A3, A4, WO, WO4)
	{*very well usable (23%,15%,39%,54%,39%,46%), moderately usable (54%,31%,46%,31%,31%,23%),*
	satisfacory (23%,46%,15%,15%,23%,23%), unsatisfactory (0%,8%,0%,0%,8%,8%)}
3.	How usable is the whole OOSP?
	{*very well usable (8%), moderately usable (62%), satisfacory (23%), unsatisfactory (8%)*}
4.	How usable is the online tutorial to OOSP?
	{*very good (8%), good (54%), satisfacory (31%), unsatisfactory (8%), I cannot evaluate. I did not need it. (0%)*}
5.	How difficult is to comprehend each test scenario? (TS1, TS2, TS3, TS4, TS5)
	{*easily comprehensible (62%,46%,0%,31%,15%), comprehensible (39%,46%,31%,62%,54%),*
	comprehensible with difficulty (0%,8%,54%,8%,23%), incomprehensible (0%,0%,15%,0%,8%)}
6.	Do you want to add a comment about OOSP?
	A: *I like the straightforward user interface, quite well arranged, no real objection.*
	A: *It would be good to have some explanatory notes for selection of ontology metrics in search approach 2.*
	A: *It would be good to better explain how searching works in search scenario 2, i.e. how searching criteria effect searched output.*
	A: *It would be good to improve the user interface. Is is not well arranged.*

usability of OOSP (Questions 2, 3 and 4) and comprehensibility of TSs (Question 5). The open Question 6 asks users for any additional comments about OOSP provided the user wanted to complement his/her answers in the questionnaire.

Discussion of Users' Experiment Results I present my summarized conclusions based on the conducted experiment, while the detailed results are available on a website, `https://owl.vse.cz/OOSP-article/`.

Regarding the usability and comprehensibility in practice, each TS is analyzed according to a ratio of users who managed to successfully fulfill the TS, which means a proper choice of the search approach and search restrictions. The most successful tasks were TS 1 and TS 2 where 85% (11 of 13) users succeeded. Almost 70% (9 of 13) managed successfully done TS 4 and 62% (8 of 13) users accomplished TS 5. The worst results users achieved in the case of TS 3 where 31% (4 of 13) users succeeded. While TS 1 was "warming up" TS where users could follow detail information, lexical based TS 2 and TS 4 were already described in more abstract way. Their difficulty was alleviated by the lower number of parameters to set up. TS 5 achieved a relatively high success ratio (62%) although this TS was difficult to comprehend (see below) and number of parameters was higher. Bad results in TS 3 can be attributed to arduous description of the assignnment where proper names of ontology metrics metadata were missing. All other users (9 of 13) managed to properly opted at least some required ontology metrics.

Regarding the time efficiency (only successful cases are considered), the fastest TSs were lexical based ones where users needed 116 and 156 seconds for TS 2 and TS 4 on average. Although TS 5 was difficult to proceed, users needed only 265 seconds on average. TSs dealing with Approach 1 were the most time consuming because users has to cope with selection from many ontology metrics metadata. TS 1 took 355 seconds and TS 3 1037 seconds on average. The interaction of the users with OOSP lasted one hour, during which time OOSP smoothly managed to cope with all users' requests without any problems. It was run on one-core Debian server with 18 GB RAM. The users did not encounter any issues with OOSP but I got several suggestions (as stated in Question 6 in Table 4.2) for improving the user interface during the testing.

Based on these numbers, we can conclude that the users did not generally have a problem coping with OOSP and comprehending the user interface of OOSP and the assignment of TSs. However, TSs with different difficulty levels disclosed that OOSP should rather be used by well-trained users. Especially in the case of the first search approach, a user can benefit from his or her experience with the names and meanings of different ontology metrics. This has been shown by comparing the results of TS 1 and TS 3. While in the case of TS 1 users could follow concrete instructions about names of the ontology metrics, in the case of TS 3 users were hesitant about which names of the ontology metrics metadata correspond to the specified requirements. This could be alleviated by improving navigation within ontology metrics, potentially with a look-up service based on the ontology metrics descriptions.

Discussion of the Questionnaire Results Regarding the questionnaire, answers to Question 1 indicate that users think that all Approaches are useful where Approach 1 is the most useful (100% of users think it is at least partly useful) and the Approach 4 consider as the least useful (92% users think it is at least slightly useful). Further, all Approaches and the options for working with the output ontologies are considered at least moderately usable by majority of users except Approach 2 where 46% users consider it as only satisfactory.

Further analyzing answers for related approaches I find that 60% of users, who found Approach 3 very useful, consider Approach 4 as at least partly useful. In the case of Approach 1 if users considered it as very useful, 83% of them found Approach 2 at least partly useful. Additionally, all users, who considered Approach 2 very useful, found Approach 1 at least partly useful. Regarding usefulness and usability, all users, who considered Approach 1, 3 and 4 (separately) as very useful, found their means of specification of individual search restrictions as at least moderately usable. Considering services for working with found ontology corpus all users, who found Approach 1 and 4 (separately) as very useful, found their respective services at least moderately usable. In the case of Approach 3, 60% of users who found the approach very useful considered respective services at least moderately usable. By aggregating answers for each user and his or her answers to six particular questions dealing with usability I found that 62% of users answered the overall question about usability of the whole OOSP in agreement to the aggregated result. 80% of users, who answered to the overall question differently, downgraded their answers with regard to their answer which would be based on average.

Further, 69% of users found the whole OOSP as very or well moderately usable. Only one user found the OOSP unsatisfactory regarding usability. Similarly for the online tutorial, only one user found the tutorial unsatisfactory. Regarding assessment of difficultness of TSs, the most comprehensible was TS 1, and then TS 2 and TS 4. The most difficult was TS 3 where 54% users found it comprehensible with difficulty and 15% incomprehensible. Based on the provided feedback from the users, this was mainly due to an ambiguous formulation of the requirements in Czech. Finally, TS 5 was considered as comprehensible with difficulty or incomprehensible by 30% users.

Based on the results of the questionnaire, we can conclude that (1) users generally tend to consider the search approaches useful means for the ontology benchmark construction; and (2) OOSP is considered usable by majority of the users. The practical results achieved by each user fulfilling each TS correspond to their answers in the questionnaire.

4.5 Related Work

Regarding a generation of synthetic ontologies, [50] recently introduced *OntoBench* which was already discussed in Section 2.4. OntoBench and OOSP complement each other in providing different services to the ontology benchmark construction. It could be even interesting to combine OntoBench and OOSP so that OOSP would provide its search approaches for a collection of synthetic ontologies generated by OntoBench.

Regarding the ontology benchmark construction by searching of existing ontologies, the most relevant is the work presented by [53] where the Manchester OWL repository is introduced. It contains a crawl-based Manchester OWL Corpus (MOWLCorp), and a snapshot of BioPortal and Oxford Ontology Library. The goal of this repository is to create and share ontology datasets. It provides access to six pre-constructed datasets and an experimental REST-based web service that should allow users to create a custom dataset.

[53] also mentioned an experimental data set creator allowing users to create custom datasets based on a wide range of ontology metrics metadata. However, on the respective website[15] there is only available offline generation of custom datasets where a user can specify his/her requirements: ontology pool, import handling, OWL2 profiles and special wishes specified in a HTML form, while the custom dataset is to be generated offline by the portal maintainers.

In comparison, my work focuses on the web-based front-end allowing to build an experimental ontology set useful as a benchmark for ontology tool developers and ontology experimenters. Therefore, I do not precompile any ontology set collections but I rather provide distinct ontology search approaches including a broad range of ontology metrics that can work as on-the-fly restrictions. Besides the BioPortal repository, I also considered the LOV repository, NanJing repository and OntoFarm collection since they belong to the most famous ontology repositories nowadays. To cover potentially many cases of different use, I also provide extra metric types, such as taxonomy, annotation, and detail ontology metrics metadata. Further, I put more emphasis on different types of additional downloads: besides actual ontologies (and optionally their imports) it is also possible to download a table with the ontology metric values and summary statistics, plus the associated graphs. While I concentrate more on ontology benchmark construction on the fly using different search approaches, [53] concentrate more on the sharing aspect. Each of the six pre-constructed datasets has its own unique ID, according to which a user can download it.

[15] `http://mowlrepo.cs.manchester.ac.uk/generate-custom-dataset/`

4.6 Chapter Summary

This chapter presents OOSP, a web-based tool allowing ontology developers and experimenters to create an ontology benchmark based on a selected search approach and ontology pool. OOSP provides four different ontology search approaches and a high number of different ontology metrics metadata, which can be useful for various use cases (e.g., benchmarking ontology repair tools, ontology visualization tools or reasoners) and scenarios (e.g., a priori known specific requirements in terms of ontology metrics or a request for ontologies similar to a sample one). Further, this chapter contains a summary of the insightful feedback from an expert interview and the report about the performed experiment with users following test scenarios focusing on usability, usefulness and comprehensibility.

Within search Approach 1 a user can ask for a randomly selected subset of k ontologies. In the future, we will consider providing stratified sub-sampling as suggested by the expert. Further, on the one hand, random sub-sampling can make benchmarking more reliable since ontologies are randomly selected but, on the other hand, the selected subset can include near-duplicates or outliers, and such subset would not be representative enough. There are basically three meaningful solutions: *homogeneous*, *distinct* or *representative*. A homogeneous benchmark would have k ontologies with as similar metric values as possible. A distinct benchmark would aim at k ontologies with as dissimilar metric values as possible, thus covering outliers. Finally, a representative benchmark should represent the space of all existing metric values as well as possible. In the future, we plan to support all these variants, i.e., a user could ask not only for a random selection of k ontologies but also for a homogeneous, distinct or representative selection of k ontologies. While homogeneous and representative selections could be solved by clustering approaches (introduced, e.g., by [36]), the distinct selection could be solved by the diversity heuristics presented by [20].

In line with the expert's opinion from the interview, we plan to investigate how to achieve nearly up-to-date snapshots, e.g., from BioPortal and LOV. While this is technically feasible for LOV, storing snapshots of BioPortal is space-demanding due to very large ontologies. We also further plan to support better reproducibility of the constructed ontology benchmarks in the future by sharing them on a dedicated website within OOSP. However, since this service could be space-demanding, instead of permanent storage of each ontology set, we would rather enable an option to store the setting of each ontology benchmark construction. In fact, this service is already internally available, but it needs addition of a mechanism which would ask a user for a permission to make it publicly shareable, and ideally also for information about the purpose of the given ontology benchmarking corpus. Finally, we plan to provide a REST interface for accessing the functions available in OOSP.

With regard to the overview of ontology benchmarks given in Chapter 2 OOSP has its potential to help people completing existing benchmarks or make new benchmarks based on ontologies found in OOSP.

Chapter 5

Benchmarks in Action

Ontology tool developers can apply the ontology benchmarks manually or they can use some automatic or semi-automatic environment. While the evaluation scheme can differ with regard to the ontology tool category, there are several projects aiming at an automatic support of ontology tool benchmarking. The SEALS (Semantic Evaluation at Large Scale) project[1] built a platform for an automatic evaluation at large scale. The benefits of automation from the SEALS perspective, as described in [23], were as follows:

- It provides automatic and uniform results.
- It can assess portability of systems since they are not run by systems' developers.
- The same controlled environment allows for measuring performance not only in terms of precision and recall but also in terms of speed, network consumption, memory and scalability.
- New tests can be added and run on existing systems.
- The tests can be re-run and results are archived. This enables reproducibility.

[1] `http://www.seals-project.eu/`

The SEALS platform was described in [27]. It follows service-oriented approach to store and process benchmarking resources. It contains a number of components:

- *SEALS portal* provides an HTML interface for interacting with the SEALS platform.
- *SEALS service manager* represents the core module of the platform which provides programmatic interfaces for services of the SEALS platform.
- *SEALS repositories* allow managing various entities used within the platform, e.g., benchmarks, tools, evaluation descriptions and results.
- *Runtime evaluation service* allows benchmarking of a certain tool using some specific benchmark.

The SEALS platform was used in many evaluation campaigns from 2011 till 2015. For example, in OAEI 2011.5[2] participants had to register their tool in the platform, wrap (by implementing required interfaces) and upload their tools according to provided tutorial. Organizers had to prepare benchmark description[3] and upload their benchmark to the SEALS platform.

In 2016 the SEALS platform has been disabled and the infrastructure was downgraded to the repositories with benchmarks and the SEALS client for running benchmarked ontology tool. Uploading of wrapped benchmarked tools to the SEALS platform was not available any more. For example, in OAEI 2016 participants had to upload their tools via Google form. For the OAEI organizers this situation was an incentive to replace incomplete SEALS infrastructure by a new HOBBIT platform.

The HOBBIT (Holistic Benchmarking of Big Linked Data) project[4] aims at benchmarking of big linked data, however it could also be used for benchmarking of some ontology

[2] `http://oaei.ontologymatching.org/2011.5/seals-eval.html`

[3] E.g., the benchmark description based on OntoFarm is at `http://repositories.seals-project.eu/tdrs/testdata/persistent/conference/conference-v1/suite/`

[4] `https://project-hobbit.eu/`

tool categories mentioned in this thesis. One benefit in comparison with SEALS is that system developers could run benchmarking tests by themselves. The HOBBIT platform, as described in [61] contains several components which are implemented as docker containers and which communicate to each other using RabbitMQ[5] as a message bus. There are two types of components: *platform components* that are always running and all components that belong to a certain experiment such as *benchmark components* and *benchmarked system component*. The platform components are as follows:

- *Platform controller* is the central component of the HOBBIT platform which coordinates the interaction of other components, e.g., handling requests from the front-end, the starting and stopping of benchmarks etc.

- *Storage* contains the experiment results.

- *Front-end* handles the interaction with the user. It provides different functionalities for different types of users, e.g., a guest, a benchmark provider etc.

- *Analysis* is started after an experiment has been finished. It should enhance the benchmark results for which it uses the HOBBIT ontology[6] and decides which analysis it should run.

- *Logging* is used to collect the log message from the components.

The benchmark components includes the following:

- *Benchmark controller* is the central component for the experiment. It creates and controls the other benchmark components described below.

- *Data Generator* prepares the data needed for the evaluation.

[5] https://www.rabbitmq.com/

[6] https://github.com/hobbit-project/ontology/blob/master/ontology.ttl

- *Task Generator* gets the data from the data generator and generates tasks identifiable with an ID.

- *Evaluation Storage* stores the gold standard and the generated results from the benchmarked system. It sends this data to the evaluation module.

- *Evaluation Module* evaluates the results generated by the benchmarked system.

The platform can be installed and run locally. However, the HOBBIT project provides an online instance[7] where one can upload a benchmark/system, run a benchmark/challenge or check results of experiments. In the period of early November 2017 to May 2018 we were preparing the OAEI benchmarks, as the OAEI 2017.5 campaign,[8] for the HOBBIT platform. While basic functionality was working well, we encountered on many difficulties dealing with OAEI benchmark specifics. We published the paper [45] describing experience of the OAEI to HOBBIT migration at the Ontology Matching workshop 2018.

Regarding my conference track it was successfully prepared as a benchmark where one have to select test cases one by one; this is available as the *OAEI Conference benchmark* in the public HOBBIT platform. However, since we expect to evaluate all test cases (21) for each matching system, it also needs to enable running of all test cases at once. I was preparing this in the public HOBBIT platform named as the *OAEI Conference benchmark ALL*, but this benchmark was not successfully done. The idea behind[9] was that first all test cases are prepared as tasks in an array of tasks with all information such as a source, a target and reference files. Additionally, there was added a queue name for each task to enable distinguishing between different tasks. Later on, tasks are sent to the task system adapter one by one. However, the system was always only working with ontologies from one task. This is still an open issue for future.

[7] `http://master.project-hobbit.eu/`

[8] `http://oaei.ontologymatching.org/2017.5/`

[9] The source code is at `https://gitlab.com/OndrejZamazal/conference`

The conference track evaluates alignments of system matchers using different evaluation methods. Although, the evaluation could be in principle implemented in the HOBBIT platform, for OAEI purposes it is also useful to have the alignments of system matchers available for each task. Since there is no direct support in HOBBIT for this, I tried to store the results onto some external host. I tested saving alignment onto Dropbox[10] but the communication between Dropbox and HOBBIT was not successful. Saving a file externally also failed for the Google Drive[11] (due to library dependency conflicts) and FTP (it was unable to communicate with FTP server). Finally, I overcome this by directly exporting alignments of system matchers into a log for corresponding experiment. While this solution works, it is certainly just ad-hoc solution. Thus, this is also an open issue for future.

Since the other track organizers and system developers had similar issues, we requested improvements in HOBBIT before OAEI can smoothly migrate. It was decided that the OAEI 2018 campaign[12] will be mostly run on SEALS while only several benchmarks will be operated on HOBBIT.

[10] https://www.dropbox.com/

[11] http://drive.google.com/

[12] http://oaei.ontologymatching.org/2018/

Chapter 6

Summary and Prospects

This habilitation thesis presents an overview of ontology benchmarks as an important part of the semantic web. While the overview is based on the proposed ontology tool categorization enabling a basic grouping of ontology benchmarks, ontology benchmarks are further described using the activities they support along with their characteristics. Based on the description of ontology benchmarks, I automated a generation of knowledge base for a NEST-based recommender and provided its simple web-based interface. Thanks to the fact that the knowledge base is generated, additional tuning of rules and their weights is possible via meta-rules. Further, it is possible to change the description of the current benchmarks or to add missing ontology benchmarks into the recommender via the input table. An alternative approach to create an ontology benchmark recommendation could be based directly on a required ontology tool category from which the tool should be benchmarked. However, since the description of each ontology tool category is always, to a certain extent, simplified, e.g., as was done for the capturing of typical characteristics of ontology tool categories in Table 1.2, I consider it better to base the recommendation, on the one side, on particular required activities and characteristics for given benchmarking need and, on the other side, on a description of specific ontology benchmarks. The validation of this is left for future research.

Furthermore, I presented in more detail my two contributions to the field of ontology benchmarks in the semantic web. They represent two different approaches to ontology benchmark construction. The first approach, the OntoFarm collection, deals with manual ontology benchmark construction and the second, the OOSP tool, enables users to automatically construct an ontology benchmark. OntoFarm has proved to be an important ontology benchmark for many researchers and in many international projects. The OOSP tool enables a community to search for ontologies potentially useful for ontology benchmarks, using different search approaches. This represents a complementary approach to manual-based and synthetic-based approaches to ontology tool benchmark construction. Because ontology benchmarking is still an open issue, there are several different aspects in which OntoFarm or OOSP could be extended or improved. According to the survey of the OntoFarm summary paper (updated in Chapter 3), the most beneficial directions for OntoFarm extensions are as follows:

- extending a number of reference alignments;
- establishing new test cases dealing with matching domain ontologies to one large ontology;
- collecting naming or trivial modeling issues.

Based on users' and experts' feedback on OOSP reflected in Chapter 4, the most beneficial directions for OOSP extensions are as follows:

- assuring up-to-date snapshots of source repositories, e.g., BioPortal, and LOV;
- supporting reproducibility of the constructed ontology benchmarks, e.g., via Zenodo.org platform;[1]
- providing a REST-based interface for accessing available services in OOSP.

[1] https://zenodo.org/

Finally, I also described my experience (Chapter 5) with two environments for an automatic ontology tool benchmarking. As it transpired, this is also an open issue for future research. It is supposed that all these issues should be inspected in collaboration with other researchers from abroad and/or with students from our university within their Bachelor, Master or Doctoral theses.

Bibliography

[1] M. Achichi, M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, I. Harrow, V. Ivanova, et al. Results of the Ontology Alignment Evaluation Initiative 2016. In *11th ISWC workshop on ontology matching (OM)*, pages 73–129. CEUR-WS Vol-1766, 2016.

[2] D. Allemang and J. Hendler. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Elsevier, 2011. ISBN: 9780123859655.

[3] G. Antoniou, P. Groth, F. Van Harmelen, and R. Hoekstra. *A Semantic Web Primer*. MIT press, 2012. ISBN: 9780262018289.

[4] G. Antunes, M. Bakhshandeh, J. Borbinha, J. Cardoso, S. Dadashnia, C. Di Francescomarino, M. Dragoni, P. Fettke, A. Gal, C. Ghidini, et al. The Process Model Matching Contest 2015. *GI-Edition/Proceedings: Lecture notes in informatics*, 248:127–155, 2015.

[5] S. Bail, B. Parsia, and U. Sattler. JustBench: A Framework for OWL Benchmarking. In *International Semantic Web Conference*, pages 32–47. Springer, 2010.

[6] P. Berka. NEST: A Compositional Approach to Rule-Based and Case-Based Reasoning. *Advances in Artificial Intelligence*, 2011:4, 2011.

[7] C. Bizer and A. Schultz. The Berlin SPARQL Benchmark. *International Journal of Semantic Web and Information Systems*, 5(2):1–24, 2009.

[8] D. Brickley, R. V. Guha, and B. McBride. RDF Schema 1.1. *W3C recommendation*, 25:2004–2014, 2014.

[9] J. Broekstra, A. Kampman, and F. Van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *International semantic web conference*, pages 54–68. Springer, 2002.

[10] L. Bühmann, J. Lehmann, and P. Westphal. DL-Learner—A framework for inductive learning on the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 39:15–24, 2016.

[11] J. J. Carroll and J. De Roo. OWL Web Ontology Language Test Cases. Technical report, W3C, 2004.

[12] M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, R. Granada, V. Ivanova, E. Jiménez-Ruiz, et al. Results of the Ontology Alignment Evaluation Initiative 2015. In *10th ISWC workshop on ontology matching (OM)*, pages 60–115. CEUR-WS Vol-1545, 2015.

[13] M. Cheatham and P. Hitzler. Conference v2. 0: An Uncertain Version of the OAEI Conference Benchmark. In *International Semantic Web Conference*, pages 33–48. Springer, 2014.

[14] G. Cheng, S. Gong, and Y. Qu. An Empirical Study of Vocabulary Relatedness and Its Application to Recommender Systems. In *International Semantic Web Conference (2011)*, pages 98–113. Springer, 2011.

[15] M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta. Characterizing Knowledge on the Semantic Web with Watson. In *Evaluation of Ontologies and Ontology-based tools (2007)*, 2007.

[16] L. Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs. Swoogle: A Semantic Web Search and Metadata Engine. In *the 13th ACM inter. conference on Information and knowledge management*, pages 652–659. ACM, 2004.

[17] Z. Dragisic, K. Eckert, J. Euzenat, D. Faria, A. Ferrara, R. Granada, V. Ivanova, E. Jiménez-Ruiz, A. O. Kempf, P. Lambrix, et al. Results of the ontology alignment evaluation initiative 2014. In *Proceedings of the 9th International Conference on Ontology Matching-Volume 1317*, pages 61–104. CEUR-WS Vol-1317, 2014.

[18] Z. Dragisic, V. Ivanova, P. Lambrix, D. Faria, E. Jiménez-Ruiz, and C. Pesquita. User validation in ontology alignment. In *International Semantic Web Conference*, pages 200–217. Springer, 2016.

[19] Z. Dragisic, V. Ivanova, H. Li, and P. Lambrix. Experiences from the anatomy track in the ontology alignment evaluation initiative. *Journal of biomedical semantics*, 8(1):56, 2017.

[20] M. Drosou and E. Pitoura. Comparing Diversity Heuristics. Technical report, Computer Science Department, University of Ioannina, 2009.

[21] M. Dudáš, S. Lohmann, V. Svátek, and D. Pavlov. Ontology Visualization Methods and Tools: A Survey of the State of the Art. *The Knowledge Engineering Review*, 33, 2018.

[22] Q. Elhaik, M.-C. Rousset, and B. Ycart. Generating Random Benchmarks for Description Logics. In *Description Logics*, 1998.

[23] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, 2013. ISBN: 9783642387203.

[24] A. Ferrara, D. Lorusso, S. Montanelli, and G. Varese. Towards a Benchmark for Instance Matching. In *Ontology Matching*. CEUR-WS Vol-431, 2008.

[25] A. Ferrara, S. Montanelli, J. Noessner, and H. Stuckenschmidt. Benchmarking Matching Applications on the Semantic Web. In *The Semanic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011*, pages 108–122. Springer, 2011.

[26] R. García-Castro. *Benchmarking Semantic Web technology*, volume 3. IOS Press, 2009. ISBN: 9781607500537.

[27] R. García-Castro, M. Yatskevich, C. T. dos Santos, S. Wrigley, L. Cabral, L. Nixon, and O. Zamazal. The state of semantic technology today – overview of the first seals evaluation campaigns. 2011. Deliverable D4.2.2. Available from `http://staffwww.dcs.shef.ac.uk/people/S.Wrigley/pdf/seals-whitepaper01-final.pdf` [accessed 9 February 2018].

[28] M. Giese, A. Soylu, G. Vega-Gorgojo, A. Waaler, P. Haase, E. Jiménez-Ruiz, D. Lanti, M. Rezk, G. Xiao, Ö. Özçep, et al. Optique: Zooming in on Big Data. *Computer*, 48(3):60–67, 2015.

[29] B. Glimm, I. Horrocks, B. Motik, R. Shearer, and G. Stoilos. A novel approach to ontology classification. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:84–101, 2012.

[30] O. Görlitz, M. Thimm, and S. Staab. SPLODGE: Systematic Generation of SPARQL Benchmark Queries for Linked Open Data. In *International Semantic Web Conference*, pages 116–132. Springer, 2012.

[31] J. Grant and D. Beckett. RDF Test Cases. Technical report, W3C, 2004.

[32] Y. Guo, Z. Pan, and J. Heflin. LUBM: A Benchmark for OWL Knowledge Base Systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):158–182, 2005.

[33] F. Haag, S. Lohmann, S. Negru, and T. Ertl. OntoViBe 2: Advancing the Ontology Visualization Benchmark. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 83–98. Springer, 2014.

[34] F. Haag, S. Lohmann, S. Negru, and T. Ertl. OntoViBe: An Ontology Visualization Benchmark. In *VISUAL@ EKAW*, pages 14–27. Citeseer, 2014.

[35] P. Hájek. Combining functions for certainty degrees in consulting systems. *International Journal of Man-Machine Studies*, 22(1):59–76, 1985.

[36] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Elsevier, 2011. ISBN: 9789380931913.

[37] I. Harrow, E. Jimenez-Ruiz, A. Splendiani, M. Romacker, S. Negru, P. Woollard, S. Markel, Y. Alam-Faruque, M. Koch, E. Younesi, et al. Introducing the Disease and Phenotype OAEI Track. In *Ontology Matching Workshop*. CEUR-WS Vol-1766, 2016.

[38] I. Harrow, E. Jiménez-Ruiz, A. Splendiani, M. Romacker, P. Woollard, S. Markel, Y. Alam-Faruque, M. Koch, J. Malone, and A. Waaler. Matching disease and phenotype ontologies in the ontology alignment evaluation initiative. *Journal of biomedical semantics*, 8(1):55, 2017.

[39] S. Hawke and B. Parsia. OWL 2 Web Ontology Language Conformance. Technical report, W3C, 2009.

[40] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. OWL 2 web ontology language primer. *W3C recommendation*, 27(1):123, 2009.

[41] I. Horrocks and P. F. Patel-Schneider. DL systems comparison. In *Proc. of the 1998 Description Logic Workshop (DL'98)*, volume 11, pages 55–57, 1998.

[42] E. Jiménez, C. Meilicke, B. C. Grau, I. Horrocks, et al. Evaluating Mapping Repair Systems with Large Biomedical Ontologies. In *26th International workshop on description logics*. CEUR-WS Vol-1014, 2013.

[43] E. Jiménez-Ruiz and B. C. Grau. Logmap: Logic-Based and Scalable Ontology Matching. In *International Semantic Web Conference*, pages 273–288. Springer, 2011.

[44] E. Jiménez-Ruiz, B. C. Grau, I. Horrocks, and R. Berlanga. Logic-based assessment of the compatibility of UMLS ontology sources. *Journal of biomedical semantics*, 2(1):S2, 2011.

[45] E. Jiménez-Ruiz, T. Saveta, O. Zamazal, S. Hertling, M. Röder, I. Fundulaki, A.-C. Ngonga Ngomo, M. A. Sherif, et al. Introducing the HOBBIT platform into the Ontology Alignment Evaluation Campaign. In *Ontology Matching Workshop*. To appear in CEUR-WS, 2018.

[46] C. Jonquet, A. Toulet, E. Arnaud, S. Aubin, E. Dzalé-Yeumo, V. Emonet, J. Graybeal, M. A. Musen, C. Pommier, and P. Larmande. Reusing the NCBO BioPortal technology for agronomy to build AgroPortal. In *Inter. Conference on Biomedical Ontologies ICBO'16*, page 3, 2016.

[47] A. Karmacharya, C. Cruz, and F. Boochs. Spatialization of the Semantic Web. *Semantics-Advances in Theories and Mathematical Models*, 2012.

[48] A. Khiat, M. Benaissa, and E. Jiménez-Ruiz. ADOM: arabic dataset for evaluating arabic and cross-lingual ontology alignment systems. In *Proceedings of the 10th International Workshop on Ontology Matching ISWC*. CEUR-WS Vol-1545, 2015.

[49] R. Likert. A technique for the measurement of attitudes. *Archives of psychology*, pages 5–55, 1932.

[50] V. Link, S. Lohmann, and F. Haag. OntoBench: Generating Custom OWL 2 Benchmark Ontologies. In *International Semantic Web Conference*, pages 122–130. Springer, 2016.

[51] S. Lohmann, S. Negru, F. Haag, and T. Ertl. Visualizing ontologies with VOWL. *Semantic Web*, 7(4):399–419, 2016.

[52] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan, and S. Liu. Towards a complete OWL ontology benchmark. In *European Semantic Web Conference*, pages 125–139. Springer, 2006.

[53] N. Matentzoglu, D. Tang, B. Parsia, and U. Sattler. The Manchester OWL Repository: System Description. In *Proceedings of the 2014 International Conference on Posters & Demonstrations*, pages 285–288. CEUR-WS Vol-1272, 2014.

[54] C. Meilicke, R. García-Castro, F. Freitas, W. R. Van Hage, E. Montiel-Ponsoda, R. R. De Azevedo, H. Stuckenschmidt, O. Šváb-Zamazal, V. Svátek, A. Tamilin, et al. MultiFarm: A Benchmark for Multilingual Ontology Matching. *Web Semantics: Science, Services and Agents on the World Wide Web*, 15:62–68, 2012.

[55] C. Meilicke, H. Stuckenschmidt, and O. Šváb-Zamazal. A Reasoning-Based Support Tool for Ontology Mapping Evaluation. *The Semantic Web: Research and Applications*, pages 878–882, 2009.

[56] C. Meilicke, C. Trojahn, O. Šváb-Zamazal, and D. Ritze. Multilingual Ontology Matching Evaluation–A First Report on Using MultiFarm. In *Extended Semantic Web Conference*, pages 132–147. Springer, 2012.

[57] M. A. Musen. The Protégé Project: A Look Back and a Look Forward. *AI Matters*, 1(4):4–12, 2015.

[58] Z. Pan. Benchmarking DL Reasoners Using Realistic Ontologies. In *OWLED*. CEUR-WS Vol-188, 2005.

[59] B. Parsia, N. Matentzoglu, R. S. Gonçalves, B. Glimm, and A. Steigmiller. The OWL Reasoner Evaluation (ORE) 2015 Competition Report. In *International Semantic Web Conference*, pages 159–167. Springer, 2016.

[60] C. Pesquita, D. Faria, E. Santos, and F. M. Couto. To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In *Proceedings of the 8th International Conference on Ontology Matching*, pages 13–24. CEUR-WS Vol-1111, 2013.

[61] M. Röder, A.-C. Ngonga Ngomo, and S. Martin. Detailed Architecture of the HOBBIT Platform. 2015. Deliverable D1. Available from `https://project-hobbit.eu/wp-content/uploads/2016/11/D2.1_Detailed_Architecture_of_the_HOBBIT_Platform.pdf` [accessed 12 February 2018].

[62] M. Schmidt, O. Görlitz, P. Haase, G. Ladwig, A. Schwarte, and T. Tran. Fedbench: A Benchmark Suite for Federated Semantic Data Query Processing. In *International Semantic Web Conference*, pages 585–600. Springer, 2011.

[63] A. Solimando, E. Jiménez-Ruiz, and G. Guerrini. Pushing the Limits of OWL 2 Reasoners in Ontology Alignment Repair Problems. *Intelligenza Artificiale*, 10(1):1–18, 2016.

[64] A. Solimando, E. Jimenez-Ruiz, and G. Guerrini. Minimizing conservativity violations in ontology alignments: algorithms and evaluation. *Knowledge and Information Systems*, pages 1–45, 2017.

[65] S. Staab and R. Studer. *Handbook on ontologies.* Springer, 2013. ISBN: 9783540709992.

[66] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López. The NeOn methodology for ontology engineering. In *Ontology engineering in a networked world*, pages 9–34. Springer, 2012.

[67] O. Šváb, V. Svátek, P. Berka, D. Rak, and P. Tomášek. Ontofarm: Towards an Experimental Collection of Parallel Ontologies. Citeseer, 2005.

[68] V. Svátek, O. Zamazal, and M. Vacura. Categorization Power of Ontologies with Respect to Focus Classes. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016*. Springer, 2016.

[69] E. Thiéblin, M. Cheatham, C. Trojahn, O. Zamazal, and L. Zhou. The First Version of the OAEI Complex Alignment Benchmark. In *Poster Track of ISWC*. To appear in CEUR-WS, 2018.

[70] E. Thiéblin, O. Haemmerle, N. Hernandez, and C. Trojahn. Towards a complex alignment evaluation dataset. In *The 12th International Workshop on Ontology Matching*. CEUR-WS Vol-2032, 2017.

[71] E. Thiéblin, O. Haemmerle, N. Hernandez, and C. Trojahn. Task-Oriented Complex Ontology Alignment – Two Alignment Evaluation Sets. In *The Semanic Web: Research and Applications - 15th Extended Semantic Web Conference, ESWC 2018*. Springer, 2018.

[72] J. W. Tukey. Exploratory Data Analysis. 1977. ISBN: 9780201076165.

[73] P.-Y. Vandenbussche, G. A. Atemezing, M. Poveda-Villalón, and B. Vatant. Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web. *Semantic Web*, (Preprint):1–16, 2015.

[74] S.-Y. Wang, Y. Guo, A. Qasem, and J. Heflin. Rapid Benchmarking for Semantic Web Knowledge Base Systems. In *International Semantic Web Conference*, pages 758–772. Springer, 2005.

[75] A. R. Weiss. Dhrystone Benchmark: History, Analysis, Scores and Recommendations. Citeseer, 2002.

[76] T. Weithöner, T. Liebig, M. Luther, and S. Böhm. What's wrong with OWL benchmarks. In *Proc. of the Second Int. Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006)*, pages 101–114. Citeseer, 2006.

[77] P. L. Whetzel, N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, and M. A. Musen. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic acids research*, 39(suppl 2):541–545, 2011.

[78] M. Yatskevich. A Large Scale Dataset for the Evaluation of Ontology Matching Systems. *The Knowledge Engineering Review Journal*, 24, 2009.

[79] O. Zamazal, M. Dudáš, and V. Svátek. Augmenting the Ontology Visualization Tool Recommender: Input Pre-Filling and Integration with the OOSP Ontological Benchmark Builder. In *SEMANTiCS (Posters, Demos, SuCCESS)*. CEUR-WS Vol-1695, 2016.

[80] O. Zamazal and V. Svátek. OOSP: Ontological Benchmarks Made on the Fly. In *1st Inter. Work. on Summarizing and Presenting Entities and Ontologies (2015)*. CEUR-WS Vol-1556, 2015.

[81] O. Zamazal and V. Svátek. Patomat-Versatile Framework for Pattern-Based Ontology Transformation. *Computing and Informatics*, 34(2):305–336, 2015.

[82] O. Zamazal and V. Svátek. Facilitating Ontological Benchmark Construction Using Similarity Computation and Formal Concept Analysis. In *SEMANTiCS 2016 (demo)*. CEUR-WS Vol-1695, 2016.

[83] O. Zamazal and V. Svátek. Ontology Search by Categorization Power. In *2nd International Workshop on Summarizing and Presenting Entities and Ontologies (2016)*. CEUR-WS Vol-1605, 2016.

[84] O. Zamazal and V. Svátek. The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2017.

List of Figures

List of Tables