PhD-FSTM-2020-21
The Faculty of Sciences, Technology and Medicine

DISSERTATION

Defence held on 15/06/2020 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Niklas KOLBE
Born on 6 February 1990 in Hamburg (Germany)

A Formal Approach to Ontology Recommendation for Enhanced Interoperability in Open IoT Ecosystems

Dissertation defence committee

Dr Yves Le Traon, dissertation supervisor
Professor, Université du Luxembourg

Dr Payam Barnaghi
Professor, University of Surrey

Dr Martin Theobald, Chairman
Professor, Université du Luxembourg

Dr Arkady Zaslavsky
Professor, Deakin University

Dr Sylvain Kubler, Vice Chairman
Associate Professor, Université de Lorraine

Abstract

The vision of the Internet of Things (IoT) promises novel, intelligent applications to improve services across all industries and domains. Efficient data and service discovery are crucial to unfold the potential value of cross-domain IoT applications. Today, the Web is the primary enabler for integrating data from distributed networks, with more and more sensors and IoT gateways connected to the Web. However, semantic data models, standards and vocabularies used by IoT vendors and service providers are highly heterogeneous, which makes data discovery and integration a challenging task.

Industrial and academic research initiatives increasingly rely on Semantic Web technologies to tackle this challenge. Ongoing research efforts emphasize the development of formal ontologies for the description of Things, sensor networks, IoT services and domain-dependent observations to annotate and link data on the Web. Within this context, there is a research gap in investigating and proposing ontology recommendation approaches that foster the reuse of most suitable ontologies relevant to semantically annotate IoT data sources. Improved ontology reuse in the IoT enhances semantic interoperability and thus facilitates the development of more intelligent and context-aware systems. In this dissertation, we show that ontology recommendation can form a key building block to achieve this consensus in the IoT. In particular, we consider large-scale IoT systems, also referred to as IoT ecosystems, in which a wide range of stakeholders and service providers have to cooperate. In such ecosystems, semantic interoperability can only be efficiently achieved when a high degree of consensus on relevant ontologies among data providers and consumers exists.

This dissertation includes the following contributions. First, we conceptualize the task of ontology recommendation and evaluate existing approaches with regard to IoT ecosystem requirements. We identify several limitations in ontology recommendation, especially concerning the IoT, which motivates the main focus on ontology ranking in this dissertation. Second, we subsequently propose a novel approach to ontology ranking that offers a fairer scoring of ontologies if their popularity is unknown and thus helps in providing a better recommendation in the current state of the IoT. We employ a 'learning to rank' approach to show that qualitative ranking features can improve the ranking performance and potentially substitute an explicit popularity feature. Third, we propose a novel ontology ranking evaluation benchmark to address the lack of comparison studies for ontology ranking approaches as a general issue in the Semantic Web. We develop a large, representative evaluation dataset that we derive from the collected user click logs of the Linked Open Vocabularies (LOV) platform. It is the first dataset of its kind that is capable of comparing learned ontology ranking models as proposed in the literature under real-world constraints. Fourth, we present an IoT ecosystem application to support data providers in semantically annotating IoT data streams with integrated ontology term recommendation and perform an evaluation based on a smart parking use case.

In summary, this dissertation presents the advancements of the state-of-the-art in the design of ontology recommendation and its role for establishing and maintaining semantic interoperability in highly heterogeneous and evolving ecosystems of inter-related IoT services. Our experiments show that ontology ranking features that are well designed with regard to the underlying ontology collection and respective user behavior can significantly improve the ranking quality and, thus, the overall recommendation capabilities of related tools.

Keywords: internet of things, semantic interoperability, ontology ranking, web data ecosystem, semantic annotation, learning to rank.

Acknowledgments

This work was funded by the European Union's Horizon 2020 Research and Innovation programme under grant agreement no. 688203 – bIoTope.

I would like to express my sincere thanks to my supervisor Prof. Dr. Yves Le Traon for the opportunity to pursue my doctoral studies at the University of Luxembourg. Similarly, I am thankful for the advise of Prof. Dr. Sylvain Kubler and Prof. Dr. Arkady Zaslavsky. Their guidance and feedback shaped the path that led to this dissertation and taught me how to write scientific papers.

During my doctoral studies, I have been lucky to meet many inspiring, supportive and outstanding people that strongly influenced my work. I would like to express my gratitude to two researchers that completely changed my research journey. First, my special thanks goes to Dr. Pierre-Yves Vandenbussche. Your positive response to my out-of-the-blue request resulted in a successful collaboration that forms the core contribution of this dissertation. Our technical discussions boosted my progress and steered my work in the right direction and without your support this dissertation would obviously not be the same. Second, I would like to thank Dr. Simon Cox for inviting me to the 2019 Dagstuhl workshop on metadata interoperability. The event was a unique opportunity for me as a relatively unexperienced researcher to be exposed to a diverse group of leading experts working on semantic interoperability. This experience allowed me to understand the limitations of my research and I believe that these reflections completely changed my standings as a researcher.

I am grateful to the dissertation committee, among them not mentioned so far, Prof. Dr. Martin Theobald for chairing the committee and Prof. Dr. Payam Barnaghi for his detailed, insightful comments and helpful suggestions to improve this dissertation.

I would also like to thank my current and former colleagues at the university for the entertaining breaks and the opportunity to learn about completely different research streams. In particular, I would like to thank Dr. Ludovic Mouline for his friendship, countless discussions and continuous feedback. Also, I would like to thank Dr. Marinos Kintis for helping me to better understand the world of academic research and Nikolaos Antoniadis for his companionship.

Lastly, I would like to express my heartfelt thanks to my family and friends for taking part in my doctoral journey. Their continuous support played a key role for me to successfully complete these studies. Thanks to my partner, Anar, without whom my life would not be what I would want it to be.

Luxembourg, June 2020

Niklas Kolbe

Contents

List of Abbreviations

ABox Assertional Box. 15
AHP Analytical Hierarchy Process. 31, 54
AIOTI Alliance for Internet of Things Innovation. 5
AP Average Precision. 25
API Application Programming Interface. 7, 31, 32, 34–38, 48, 49, 52, 79, 106, 108, 111, 116, 118

BTC Billion Triple Challenge. 36

CSV Comma-Separated Values. 106
CTR Click Through Rate. 41

DBN Dynamic Bayesian Network model. 90, 91
DCG Discounted Cumulative Gain. 26
DCMI Dublin Core Metadata Initiative. 75, 130
DCTR Document-based Click-Through Rate model. 90, 91
DL Description Logic. 14, 15, 32
DOL Distributed Ontology, Model and Specification Language. 38

ERR Expected Reciprocal Rank. 25–27, 77, 79
EU European Union. 128

FAIR Findable, Accessible, Interoperable and Reusable. 7, 8, 124
FOL First Order Logic. 16

HTML Hypertext Markup Language. 48, 63
HTTP Hypertext Transfer Protocol. 50, 106, 115, 117

IDF Inverse Document Frequency. 23, 53, 95
IEEE Institute of Electrical and Electronics Engineers. 5
IOF Industrial Ontologies Foundry. 66
IoT Internet of Things. 3–11, 13, 17–19, 22, 28, 37, 42, 45, 46, 49, 63–69, 71–74, 76, 78, 80, 82, 84, 102, 105, 106, 108, 109, 111, 113, 115–118, 120–124
IR Information Retrieval. 11, 13, 21–23, 25–28

JSON JavaScript Object Notation. 11, 14, 67, 106, 111, 116, 117, 122
JSON-LD JavaScript Object Notation for Linked Data. 14, 48, 108, 111

KB Knowledge Base. 15
KG Knowledge Graph. 6, 15

LCIM Levels of Conceptual Interoperability Model. 5, 6, 106, 117

LOD Linked Open Data. 15, 18, 35–39, 42, 46, 48–50, 68, 72, 73, 76, 79, 80, 82, 93, 101
LOV Linked Open Vocabularies. 10, 38, 39, 63, 67, 72, 79, 80, 82–85, 87–91, 93, 94, 97, 99–102, 106, 111, 117, 118, 121, 128
LOV4IoT Linked Open Vocabularies for the Internet of Things. 8, 67, 71, 78, 82
LTR Learning To Rank. 10, 22–24, 26, 39–42, 71, 73, 76, 77, 81, 84, 85, 87–89, 93, 97–102, 121, 128

M2M Machine to Machine. 6, 108
MAP Mean Average Precision. 25, 26, 77, 79, 99
MobiVoc Open Mobility Vocabulary. 17, 20, 22, 66, 111, 130
MQTT MQ Telemetry Transport. 106

NDCG Normalized Discounted Cumulative Gain. 25, 26, 77, 79, 84, 88, 99
NP Nondeterministic Polynomial time. 15

O-DF Open Data Format. 109, 111, 113, 116–118
O-MI Open Messaging Interface. 109, 111, 113, 115–118
ORT Ontology Recommendation Tool. 7, 19, 21, 22, 30, 38, 40, 41, 45–49, 51–54, 63–69, 106, 118, 120–122, 124
OSI Open Systems Interconnection. 5
OWL Web Ontology Language. 6, 14–17, 36, 50, 75, 77

RDF Resource Description Framework. 6, 14–17, 30, 33, 37, 38, 48, 49, 53, 92, 93, 95, 101, 106, 113, 118, 129, 131
RDF/XML XML Syntax for RDF. 48
RDFa RDF in Attributes. 48
RDFS RDF Schema. 14, 15, 17, 50, 129
REST Representational State Transfer. 52, 79, 109
RIF Rule Interchange Format. 16
RQ Research Question. 9–11, 45, 71, 83

SAREF Smart Appliance REFerence. 7
SERP Search Engine Result Page. 88, 89
SHACL Shapes Constraint Language. 16
SOLID Social Linked Data. 123
SOSA Sensor, Observation, Sample, and Actuator. 17, 19, 22, 130
SPARQL SPARQL Protocol and RDF Query Language. 16, 33–38, 40, 52, 72
SSN Semantic Sensor Network. 7, 17
SWD Semantic Web Document. 48, 49, 52, 54, 64
SWoT Semantic Web of Things. 6, 7, 66
SWRL Semantic Web Rule Language. 16

TBox Terminological Box. 15, 49
TCP Transmission Control Protocol. 115
TF Term Frequency. 23, 53
TF–IDF Term Frequency–Inverse Document Frequency. 23, 33, 52–54, 95
Turtle Terse RDF Triple Language. 14, 16, 17, 48

UBM User Browsing Model. 90, 91, 97
UI User Interface. 11, 31, 33, 34, 36–38, 52

List of Figures

List of Tables

List of Listings

Part I

Open IoT Ecosystems and Semantic Interoperability

Chapter 1

Introduction

This chapter introduces the Internet of Things (IoT) ecosystem vision and related problems that form the context of this dissertation. We then describe open research challenges with respect to the semantic interoperability issues in the broader scope of the IoT. Subsequently, we present the research questions that are addressed in this dissertation and provide an overview of the contributions of this research. Finally, we summarize the thesis outline.

Contents

1.1 Thesis Context

The Internet of Things (IoT) envisions a world in which devices with networking, sensing and actuating functions (Things) are connected through the Internet, allowing *"people and things [to be] connected anytime, anyplace, with anything and anyone ideally using any path/network and any service"* [216]. As an extension of the IoT, the Web of Things (WoT) [183, 82] describes a paradigm that leverages World Wide Web (WWW, referred to as 'the Web') standards to establish interoperability among devices from heterogeneous IoT networks that communicate with different protocols on lower layers[1]. The Web has become today's primary enabler for integrating data from distributed networks, and more and more Things, e.g., as direct endpoints and through IoT gateways, are connected to the Web. Realizing the IoT vision is foreseen to bring societal, environmental and economic opportunities in industries and cities [246, 1], mainly due to the vast amount of available data that enables humans and machines to make more informed and efficient decisions. For example, in the context of smart cities [213], access to near real-time data about drivers, traffic, road conditions, parking availability and city events allows for the design of systems and services with increasing intelligence that enables more cost-efficient and sustainable transportation, such as efficient parking guidance [207].

However, today's IoT data often remains in isolated, vendor-specific systems and clouds (so-called 'vertical silos') that do not allow for data discovery, access and reuse across systems [3]. This circumstance imposes significant limitations to the IoT vision and its opportunities and, as a recent trend, businesses, governments and innovators realized that it could become more profitable to cooperate in the scope of *IoT ecosystems* rather than innovate as individual entities [233, 47, 20, 152]. Moving away from the existing, siloed approach to one of open innovation is a fundamental step to realize the IoT vision and its potential [216]. With this development, *openness* becomes an essential characteristic of systems in a Web-based IoT context that separates its design from traditional, closed, sensor-based systems [209]. IoT ecosystems are characterized as 'open data ecosystems', for which we adopt the following definition: *"a set of networks composed by autonomous actors that directly or indirectly consume, produce or provide data and other related resources (e.g., software, services and infrastructure)"* [160]. Further, *"each actor performs one or more roles and is connected to other actors through relationships, in such a way that actors collaboration and competition promotes data ecosystem self-regulation"* [160]. The distinguishing characteristic of 'being open' means that anyone following the ecosystem's agreement can join it, which typically creates a strong dependency among the different roles, e.g., the dependence of data consumers on the metadata that is published by the data providers, resulting in further challenges related to policy, technology, organization, culture, etc. [249]. The IoT ecosystem vision and respective components are illustrated in Figure 1.1. The figure shows three ecosystem roles, including end-users who own smart objects, data analysts (startups, SMEs, etc.) who may be interested in accessing smart object-related data to deliver new services, as well as data consumers that integrate this new knowledge in their decision-making processes. Moreover, various types of incentives between these stakeholders can be imagined that could be supported by a digital marketplace acting as an IoT search engine and thus enabling multimodal registration, discovery and trading of data and services (cf. [187, 29, 169, 170]).

[1] In this dissertation, we refer to 'IoT' as an umbrella term that includes the concepts of 'WoT'.

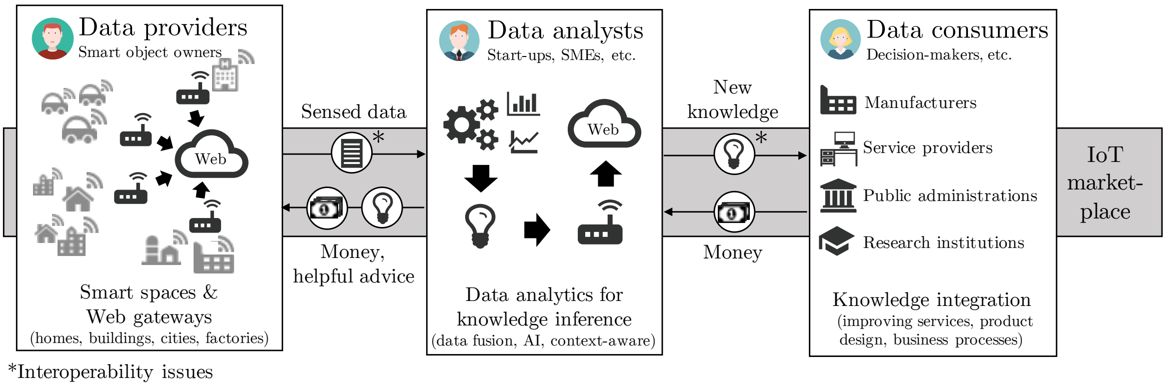


Figure 1.1: The IoT ecosystem vision (based on [169]).

One major obstacle to realizing the IoT ecosystem vision is the lack of *interoperability* among data providers and consumers caused by today's vertical silos. In this regard, several consortia and initiatives have emerged (e.g., AIOTI[2], OneM2M[3], IEEE IoT[4], Open Platform[5], IES-City[6]) that acknowledge this issue and aim for the creation of open IoT ecosystems to achieve the required seamless communication in practice. In the context of IoT ecosystems, the term 'interoperability' refers to the capability of IoT devices and services to communicate and interact with each other with minimal human intervention. In this dissertation, we follow the Levels of Conceptual Interoperability Model (LCIM) [227] as a framework to discuss interoperability issues in greater detail. LCIM is one of the first interoperability models and was previously used in the context of system-of-systems engineering and IoT [226, 184]. It describes the following six interoperability levels.

Technical. Relates to the transport layer of the Open Systems Interconnection (OSI) model, meaning there exist well-defined communication protocols and established communication infrastructure.

Syntactic. Refers to an agreement of the data structure to exchange information and a commonly agreed data format, which relates to the application layer of the OSI model.

Semantic. Describes a common understanding of the data model, the meaning of terms, relations of concepts, language, etc.

Pragmatic. Is reached when systems understand each other's interfaces, i.e., a common description of how to access systems' data.

Dynamic. Systems require the ability to track their evolution and state, including the discovery of services.

Conceptual. Requires a fully specified and unified conceptual view on the domain of discourse to be shared among all ecosystem actors.

[2]https://aioti.eu/
[3]http://www.onem2m.org/
[4]https://iot.ieee.org/
[5]https://www.opengroup.org/forum/open-platform%E2%84%A2-30-forum
[6]https://pages.nist.gov/smartcitiesarchitecture/

This dissertation focuses on challenges that arise from issues at the *semantic interoperability* level, with the goal of enhancing interoperability in the previously introduced IoT ecosystem vision. However, the LCIM model emphasizes that this is only one aspect of achieving full interoperability among systems: interoperability solutions need to fit into a broader technology stack that addresses interoperability at different levels. Semantic interoperability issues in the IoT are caused by the wide range of data modeling approaches and vocabularies used on the Web that hinder the efficient development of such cross-platform and cross-domain applications [234, 173]. The given heterogeneity of data and services makes it difficult to efficiently and on-demand discover, access and integrate relevant IoT data sources [3]. Supporting and maintaining a multitude of ever-changing standardized and non-standardized approaches to publish and consume IoT-originated data, especially in cross-domain contexts, is a problem with exponential complexity that is also referred to as the 'industrial IoT connectivity challenge' [184].

However, data heterogeneity and related semantic interoperability issues on the Web are neither novel nor exclusive to IoT ecosystem use cases. From a Web data ecosystem perspective, the data produced by IoT devices merely contribute with additional complexity, volume, dynamicity as well as the need for Machine to Machine (M2M) communication to the already existing millions of datasets and services on the Web [18]. The *Semantic Web* [23], which emerged in the early 2000s, comprises a set of technologies that complement the Web intending to make Web data machine-readable and reusable. Semantic Web technologies, such as the Resource Description Framework (RDF) [147] and Web Ontology Language (OWL) [8], present a formal framework to describe the semantics of data and fundamentally employ the Web's capability to describe the relation (link) of resources. Extendable data schemas are referred to as *ontologies* that allow data to be discoverable and reusable. The previous success of Semantic Web technologies in the biomedical domain (e.g., BioPortal [157]), enterprise data integration (based on so-called Knowledge Graphs (KGs) [159]), Web search enhancements (e.g., based on schema.org [81]) and early success in Web dataset search (e.g., Google Dataset Search [158]) have proven the real-world feasibility to establish interoperability in open data ecosystems using Semantic Web technologies. This motivates subsequent research efforts that continue to investigate the application of Semantic Web technologies in IoT systems (also referred to as the Semantic Web of Things (SWoT) paradigm [28]) to tackle semantic interoperability issues in the context of open and connected IoT systems [18].

While the Semantic Web indeed provides a machine-understandable knowledge infrastructure on the Web that can be easily integrated into existing software environments [204], the application of Semantic Web technologies also introduces new challenges and has been subject to criticism [100]. One of the major challenges is the cost caused by the additional effort and complexity of sound semantic data annotations. For example, modeling data with ontologies is not trivial, as the fundamental principle of achieving semantic interoperability between distributed systems is to reuse existing ontology terms and establish interconnectivity between them [204, 95]. In this context, various guidelines, recommendations and best practices have to be known and followed in order to apply Semantic Web technologies in a useful manner (cf. [22, 103, 96, 110, 182, 24]). A continuous consensus on the evolving relevance of ontologies eventually enhances the semantic interoperability among systems and tools have emerged to support users of the Semantic Web in their tasks and be more efficient in the discovery and reuse of existing ontologies. We use the umbrella term *Ontology Recommendation Tools (ORTs)* to refer to such tools that fundamentally aim to provide users with relevant ontologies, such as ontology repositories and ontology search engines. Apart from technical solutions, it is worth mentioning that several initiatives exist that focus on social aspects of ecosystem interoperability. A major initiative in this regard advocates making data Findable, Accessible, Interoperable and Reusable (FAIR) [238] – an ambition that aligns with the intent and design of open IoT ecosystems. Global, high-impact initiatives such as FAIR highlight the problem that data published to the Web (whether in IoT contexts or in general) is often not reusable and that making Web data actually useful requires awareness and additional effort to comply with commonly agreed guidelines. Thus, the success of the Semantic Web and its adoption relies not just on the technology stack itself, but rather on the community and its tools that enable users to work with these technologies correctly, efficiently and in a well-integrated manner of the respective processes. This circumstance also holds true for the SWoT vision, which, e.g., lead to a subsequent call for efficient tool support to allow (semi-)automated semantic data management in IoT ecosystems [203].

In this dissertation, we address technical challenges that arise from applying Semantic Web technologies in the context of IoT ecosystems with the goal of establishing and enhancing semantic interoperability. In the following section, we present a broader and general overview of existing research challenges in this scope.

1.2 Research Challenges

The realization of the IoT ecosystem vision faces various research challenges that are rooted in the given heterogeneity of today's IoT and its subsequent semantic interoperability issues. We present four high-level challenges that are considered in the literature, which are all concerned with the application of Semantic Web technologies in the IoT to achieve interoperability.

Challenge 1: Ontology engineering for IoT setups and observations. The first challenge to establish semantic interoperability in IoT ecosystems is concerned with the development of ontologies that describe sensor (network) setups, data streams, services, device capabilities, etc. Early research on the SWoT paradigm focused on this challenge, resulting in various ontologies, such as the Semantic Sensor Network (SSN) ontology[7] [53], Smart Appliance REFerence (SAREF)[8] [56], the WoT Thing Description[9], and research that aims to define IoT ecosystem Application Programming Interfaces (APIs) through ontologies, such as the BIG IoT API [30]. However, it is worth mentioning that the need for ontologies goes beyond sensor and service descriptions: sensors, e.g., make observations of the real world that also need to be formally described. This means that IoT applications almost always rely on ontologies of the respective application domain, such as health or transportation, that may or may not already have an existing community of ontology engineers prior to emerging IoT use cases. Ontology engineering is challenging since it requires a sound understanding of the domain of discourse in depth and width. The need for defining common ontologies for the IoT [73, 18, 7] as well as respectively proposed ontologies [219,

[7]https://www.w3.org/TR/vocab-ssn/
[8]http://ontology.tno.nl/saref
[9]https://www.w3.org/TR/wot-thing-description/

199, 14] are both reviewed in the literature.

Challenge 2: Mappings and alignments between ontologies/standards. With the co-existence of various ontologies and standards used in different projects, another approach essential to semantic interoperability is to create mappings of concepts from different ontologies and external terminology resources. This challenge also includes the complex task of ontology matching for ontologies of the same domain [163], which can be challenging as it may lead to logical inconsistencies in the model. Ontology mapping has been the focus in previous IoT research projects, such as the Fiesta IoT project that achieved semantic interoperability between the FIWARE and OneM2M platforms (both using different data models) [128], as well as the SPITFIRE project that was concerned with aligning IoT ontologies [174]. The need and challenge of ontology mapping for the IoT are subject in the wider literature [74, 86]. In a recent effort, a mapping ontology on top of the WoT Thing Descriptions was proposed [49].

Challenge 3: Reuse of IoT and domain ontologies. With existing and continuously evolving ontologies and mappings on the Web, semantic interoperability can only be assured when users are able to efficiently discover and select the most appropriate ones for their needs from the pool of all available ontologies. Due to the distributed and heterogeneous nature of the Semantic Web, finding and reusing ontologies is not an easy task and has been an important subject in Semantic Web research [60, 204]. Thus, it has also been a concern of IoT projects such as LOV4IoT [84]. While these projects primarily aim at building a collection of ontologies relevant for IoT application domains, the understanding of the criteria that should be taken into account when recommending one of these ontologies for reuse is relatively unexplored, particularly in IoT aspects. Providing efficient recommendations to users to reuse ontologies is a challenging but essential task to establish an ecosystem-wide consensus on common ontologies, e.g., for the annotation of IoT data. The challenge of ontology reuse in the context of semantic interoperability in the IoT is further discussed in the literature [88, 175, 15].

Challenge 4: Ontology-based data annotation and reasoning. Lastly, with one or more selected ontologies, adding semantic annotations to IoT data streams remains a non-trivial task that comes at a cost. Providing efficient tool support is thus a crucial aspect to foster adoption and sustainability of ontology-based interoperability approaches, from both the data publisher and consumer perspective [18]. Therefore, another challenge that is addressed by the research community is concerned with (semi-)automation of respective IoT ecosystem processes, such as the publication of semantically annotated IoT data as well as their discovery and integration in intelligent applications [85]. Moreover, such tools would ideally support respective interoperability guidelines (such as FAIR). The challenges of IoT data annotation [237] and semantic-based reasoning over IoT data [62] are both discussed in the literature.

In addition to the challenges outlined above, a much wider range of challenges exists in the scope of IoT. Respective surveys cover challenges from various perspectives including enabling technologies [216, 5, 11], IoT middleware [154, 185], IoT platforms

[149], standards [16], IoT data discovery and search [71], IoT data management [180], industrial IoT [55], pervasive computing [244] as well as IoT interoperability [155] and specific focus on semantic interoperability [90, 203, 35]. In the following section, we present the research questions that are addressed in this dissertation.

1.3 Research Questions

From an IoT ecosystem perspective, no single data model should be imposed at the data provider level. It is neither feasible nor manageable to create a single data model/ontology that describes all aspects of the IoT and related domains (i.e., one that would satisfy all actors) [138]. Moreover, it is not realistic to design a single universal approach to semantically annotate sensor data for IoT gateways [171]. Nevertheless, semantic annotations are a fundamental requirement to discover and integrate available IoT data in intelligent and autonomous systems, e.g., for the enablement of WoT search engines [228, 71]. This is the key aspect that motivates the investigation of the role of ontology recommendation in open IoT ecosystems to establish semantic interoperability. Thus, this dissertation primarily focuses on Challenge 3, i.e., the following Research Questions (RQs) are concerned with ontology recommendation in the context of IoT ecosystems. Unlike Challenge 1 and 2, we consider Challenge 3 to be less explored and as a missing building block for open IoT ecosystem interoperability and a supportive component for ontology engineering and mapping. We specify the following RQs to define the scope of this dissertation regarding Challenge 3:

RQ 1. What are the limitations and requirements of existing ontology recommendation tools to support semantic interoperability in IoT settings?

RQ 2. Are current ontology ranking approaches effective for IoT ontology collections?

RQ 3. Which ontology ranking criteria are important for efficient recommendations?

RQ 4. How can ontology recommendation approaches be evaluated and compared?

Moreover, we consider the integration of ontology recommendation in IoT processes, i.e., semantic data annotation. The following RQ defines the scope of this dissertation regarding Challenge 4:

RQ 5. How can ontology recommendation be integrated efficiently in the IoT data publication process?

In the next section, we present the contributions of this dissertation that answer the RQs specified above.

1.4 Contributions

In this dissertation, we argue that the efficient recommendation of existing ontologies for reuse is a key building block to enable semantic interoperability in IoT ecosystems. Overall, recommendation is meant to guide providers and consumers in finding and reusing the most suitable ontologies for their specific intent and circumstance, and support the convergence to common, evolving annotation formats in the ecosystem community. To support this vision, this thesis provides a detailed study of ontology recommendation for IoT use cases, resulting in the following contributions that address the previously introduced RQs.

A taxonomy of ontology recommendation tools and their IoT ecosystem integration. This contribution aims at the analysis of existing ontology recommendation tools and finding their potential shortcomings in supporting the IoT ecosystem vision. An extensive survey is conducted and a classification scheme is proposed that identified 20 features along eight dimensions of related approaches. In total, 40 tools introduced in the literature are evaluated based on the proposed classification scheme, allowing the identification of limitations and open challenges of ontology recommendation tools in the context of the IoT ecosystem vision (RQ 1, RQ 2). The identified limitations through this survey motivate the following contributions of this thesis. The contribution is based on the work that has been presented in the following paper:

- Niklas Kolbe, Sylvain Kubler, Jérémy Robert, Yves Le Traon, and Arkady Zaslavsky. "Linked Vocabulary Recommendation Tools for Internet of Things: A Survey". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 127. DOI: 10.1145/3284316.

Ontology ranking approach for IoT ontology collections. A significant conclusion of the survey is that it is not clear which criteria are important to take into account when recommending ontologies. One of the most adopted criteria used in existing solutions, as identified through the survey, is a *popularity* measure that represents how often an ontology has been used before and is thus more likely to be recommended for future reuse, fostering convergence to common representation of data. In this contribution, we show that the critical popularity measure, as computed in the state-of-the-art, does not hold for IoT ontology collections (RQ 2). Instead, we investigate whether the ontologies' search relevance driven by their popularity can be predicted using qualitative features, such as availability, consistency and understandability, following a Learning To Rank (LTR) approach. The results show that qualitative features help to improve retrieval performance (RQ 3), providing a fairer ranking when no explicit popularity feedback is available, such as in the current IoT domain. The contribution is based on the work that has been presented in the following paper:

- Niklas Kolbe, Sylvain Kubler, and Yves Le Traon. "Popularity-driven Ontology Ranking using Qualitative Features". In: *International Semantic Web Conference*. Springer. 2019. DOI: 10.1007/978-3-030-30793-6_19.

Ontology term ranking evaluation benchmark. The survey and the previous study further revealed difficulties in evaluating and comparing ontology ranking approaches – as a general problem in the Semantic Web and thus also for ontology recommendation approaches in IoT ecosystems. This contribution provides a benchmark for ontology term ranking that is created from logged user feedback of the Linked Open Vocabularies (LOV) [232] ontology term search platform (RQ 4). The implicit user feedback in terms of views and clicks is used to derive explicit relevance labels for query and ontology terms. The resulting dataset overcomes several limitations of the state-of-the-art, e.g., providing sufficient relevance judgments to perform an empirical evaluation of proposed ranking models using supervised machine learning techniques, i.e., LTR. Our study further provides more detailed insights on the effectiveness of ranking features based on a real-world dataset (RQ 3). In the context of this thesis, we argue that taking user feedback into account for ontology recommendation would support convergence to common ontologies continuously, meaning the recommendation is able to take the ever-evolving schemas and changing relevance for users into account. The contribution is based on the work that has been presented in the following paper:

- Niklas Kolbe, Pierre-Yves Vandenbussche, Sylvain Kubler, and Yves Le Traon. "LOVBench: Ontology Ranking Benchmark". In: *Proceedings of The Web Conference (WWW)*. ACM, 2020. DOI: 10.1145/3366423.3380245.

IoT ecosystem application to support data stream annotation. The lack of ontology recommendation's integration in IoT tools, e.g., supporting IoT data publication and consumption, was also identified through the survey. This contribution investigates the integration of ontology recommendation for semi-automated publication of semantically annotated IoT data streams (RQ 5). The contribution includes a proof-of-concept that provides a User Interface (UI) to map and annotate data with ontology terms and generates agents that transform JavaScript Object Notation (JSON) data to semantically annotated formats following IoT standards and publish the data streams through IoT gateways. The evaluation based on a smart parking use case relies on the technology stack proposed by an IoT ecosystem research project. The contribution is based on the work that has been presented in the following paper:

- Niklas Kolbe, Jérémy Robert, Sylvain Kubler, and Yves Le Traon. "PROFICIENT: Productivity Tool for Semantic Interoperability in an Open IoT Ecosystem". In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*. ACM. 2017. DOI: 10.1145/3144457.3144479.

1.5 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 presents relevant background of Semantic Web and Information Retrieval (IR). Related work for ontology recommendation is presented in Chapter 3. Chapter 4 proposes a taxonomy and derived research directions that guide the subsequent contributions in the following chapters regarding the ranking for IoT ontology collections (Chapter 5), ontology term ranking benchmarking (Chapter 6) and integration of ontology recommendation in the IoT data publication process (Chapter 7). The thesis is concluded in Chapter 8, which further discusses future research. Table 1.1 provides an overview of the chapters, addressed RQs and related publications on which the chapters are based.

Table 1.1: Overview of Thesis Structure, Research Questions and Publications

Part	Chapter	RQ 1	RQ 2	RQ 3	RQ 4	RQ 5	Ref.
I Introduction & literature	1 Introduction						
	2 Background						
	3 Related Work						
II Ontology recommendation	4 Taxonomy	✓	✓				[121]
	5 IoT ontology ranking		✓	✓			[120]
	6 Ranking benchmark			✓	✓		[124]
III Application & conclusion	7 Ecosystem application					✓	[123]
	8 Conclusion						

Chapter 2

Background

This chapter presents relevant background regarding Semantic Web and Information Retrieval (IR) technologies. First, we introduce fundamental concepts such as ontologies, practical recommendations of the WWW Consortium (W3C) (e.g., ontology and query languages), provide an illustrative example in an IoT context and present fundamentals of ontology reuse. Second, the fundamental aspects of search and ranking, such as essential ranking types and evaluation, are introduced. Subsequently, we present the case of applying ranking techniques to ontology collections and summarize the chapter.

Contents

2.1 Semantic Web

The Semantic Web, as envisioned by Tim Berners-Lee [23], defines a technology stack for formal information representation on the Web that allows encoding the semantics of data (such as the description of concepts and their relations) with the intention to make the meaning of information on the Web machine-readable. In this section, we introduce fundamental technologies and aspects such as ontology reuse.

2.1.1 Semantic Web Technologies

In this section, we introduce the underlying technology and terminology of the Semantic Web.

Unique Resource Identifier (URI). Fundamentally, the Semantic Web relies on unique identifiers similar to Uniform Resource Locators (URLs) of the Web to refer to resources such as concepts and data points.

Resource Description Framework (RDF) [147]. Defines a Web-embedded, directed labeled graph data model to represent information. RDF graphs are typically defined by a set of RDF statements, which are referred to as triples due to their notation in the form of `(subject, predicate, object)`. The `subject` and `object` represent nodes, while a `predicate` either represents an edge to describe the relation (often referred to as *property*) between two resources (all represented with URIs) or of a resource and a literal (e.g., string, number, and datatypes defined by XML Schema Definition (XSD)). RDF can be serialized in various syntaxes, including extensions of well-known ones such as eXtensible Markup Language (XML) and JSON (i.e., RDF/XML and JavaScript Object Notation for Linked Data (JSON-LD)), as well as novel formats such as Terse RDF Triple Language (Turtle).

RDFS Schema (RDFS) [144]. The first of two currently dominant Web standards and formal languages that define a vocabulary that adds well-defined semantics to the RDF data model. RDFS provides a basic vocabulary that includes the specification of class and property hierarchies as well as statements of the domain (subject) and range (object) of a property.

Web Ontology Language (OWL) [8]. The second of the two dominant formal languages extends RDFS and is far more expressive. OWL is based on Description Logics (DLs) [106] and the WWW Consortium (W3C) recommendations include several OWL variants with different expressive power that are defined by the set of supported logical constructors in the respective variant, e.g., equivalence, disjointness, cardinality, enumeration, union, and intersection. Furthermore, OWL allows imports of statements from other ontologies and to specify metadata properties. Higher expressive power, however, comes at the cost of higher complexity (see 'ontology reasoning').

Ontology [79]. Following a widely-accepted definition for the term *ontology*, we define it as *"an explicit specification of a conceptualization"* [79], and further, that ontologies describe the meaning of names used for the entities with human-readable text and define *"formal axioms that constrain the interpretation and well-formed use of these terms"*. In the Semantic Web, an ontology corresponds to the data schema definition expressed with technologies such as RDF, RDFS and OWL. A formal specification of ontologies is presented in [136].

Vocabulary. This term, sometimes also referred to as Linked Vocabulary, is often used interchangeably with the term ontology. As argued by the W3C, *"there is no clear division between what is referred to as vocabularies and ontologies"*[1]. However, it is further noted that the term ontology is often used for *"more complex"* schema definitions, whereas the term vocabulary does not focus on a *"strict formalism"*. It can be perceived that many popular and lightweight schemas (e.g., schema.org[2]) often use the broader term vocabulary instead of ontology. Thus, the preferred term to use depends on the domain and the particular use case. In this dissertation, we use the term ontology.

Term. We use the overarching expression *term* (in Semantic Web contexts) to refer to an element in an ontology that is either a class or a property.

Linked Data [25]. Refers to the best practices to publish and interlink structured data, meaning RDF-based datasets with instances and ontologies referring to other common ontologies on the Web as well as potential links on the data level. This typically concerns open datasets, thus coining the term Linked Open Data (LOD).

Knowledge Base (KB) and Knowledge Graph (KG). Both terms are often used interchangeably in the context of the Semantic Web. A KB is commonly referred to as a populated vocabulary/ontology, i.e., ontologies along with instantiations of its classes that represent data [61]. In DL terminology, a KB encompasses the Terminological Box (TBox) for the schema, and Assertional Box (ABox) for the data [75]. While the term KB is often used in contexts where reasoning plays an important role, the term KG [101], which gained popularity through a product of the same name by Google, is used in cases where the large scale of the datasets determines the associated challenges and tasks, in which often semantic- and graph-based techniques need to be applied to reconcile and enhance the data.

Ontology reasoning [108]. Ultimately, data described based on RDF, RDFS and OWL need to be accessed and integrated with processing systems. So-called reasoners prove the semantic consistency of the logical entailments defined in ontologies and ontology-based datasets and derive explicit statements from the knowledge that is implicitly defined with previous semantics. Reasoning steps are referred to as inference and are typically implemented based on the Tableaux algorithm [108]. The reasoning complexity varies depending on the underlying formalism used, e.g., existing OWL variants range from Nondeterministic Polynomial time (NP)-complete (OWL 2 RL and OWL 2 QL), EXPTIME (OWL 2 EL), and undecidable (OWL 2 Full). The development of many available reasoners is driven by the research community and the support of languages, performance and other criteria differ significantly among implementations. Reasoners form an essential building block to make the semantic information practically processable and usable. Prominent reasoner implementations include Pellet [205] and Hermit

[1] `https://www.w3.org/standards/semanticweb/ontology`
[2] `http://schema.org/`

[201]. OWL follows the so-called open-world assumption, in which the absence of a truth statement does not imply that this statement is not true.

Shapes Constraint Language (SHACL) [119]. SHACL, on the other hand, is a constraint language that follows the closed-world assumption, i.e., missing statements are considered to be not true. SHACL allows to specify so-called shapes that can be applied on RDF datasets and validate these based on the defined constraints. Even though SHACL and OWL are based on different assumptions and specified for different purposes, the expressiveness and statements largely overlap. In practice, sometimes OWL statements are simply interpreted under closed-world assumption instead of developing a corresponding SHACL specification.

Rule languages [107]. Apart from reasoners, various rule-based systems have been adopted to the RDF stack, such as Semantic Web Rule Language (SWRL) [107] that is based on First Order Logic (FOL). The W3C recommends the Rule Interchange Format (RIF) [117] as a standardized layer to exchange rules between systems. FOL rules can complement ontologies, e.g., by defining concepts with logical conditions on defined terms in the ontology. Rules can be integrated seamlessly in ontology documents, such as SWRL in Turtle syntax.

RDF query languages [177]. Another important use case is to query data expressed in RDF. SPARQL Protocol and RDF Query Language (SPARQL) forms the Semantic Web standard for this purpose. This query language meets the requirements of the underlying triple data model of RDF with a rich feature set for formulating queries. While many public SPARQL Protocol and RDF Query Language (SPARQL) endpoints are exposed to the Web, these can easily break the underlying system as the query language is very expressive. Instead, simpler graph query languages that are designed as for service interfaces can be adapted to RDF datasets, such as GraphQL [220].

Triple store [190]. Triple stores, also referred to as RDF stores or RDF databases, are optimized for the efficient storage and retrieval of RDF data. Depending on the implementation, query interfaces further allow applying reasoning and graph queries. Well-known triple store implementations include Virtuoso [68] and Stardog[3].

Semantic Web Service [140]. This concept, fundamentally based on the OWL-S ontology [140], describes the adoption of semantic technologies to Web service design. OWL-S forms the upper ontology to model the services, including aspects such as service metadata, functions, parameters, etc. Albeit being a sound concept that enables pragmatic and potentially dynamic interoperability, Semantic Web Services have not gained a lot of traction in the wider Web community so far.

The following section presents an illustrative example of working with Semantic Web technologies.

[3] `https://www.stardog.com/`

2.1.2 Ontologies and Linked Data: IoT Example

In the context of the IoT, several research directions have emerged that often rely on Semantic Web technologies, with the primary motivation being to solve semantic interoperability issues. In context-aware computing [200, 172], e.g., ontologies are often used to develop context models that describe entities and their environment. However, ontologies are also used for context reasoning to understand the system's physical surroundings with regard to completeness, accuracy, noise, etc. Semantic models may also enrich IoT data-driven analytics [137]. In typical IoT scenarios, data is not published statically but rather as continuous data streams. Thus, being able to perform event-processing and real-time reasoning on top of data streams has been subject to research projects [62]. In the following, we present an example that aims to demonstrate the nature and caveats of working with ontologies in Semantic Web and IoT settings. Moreover, it serves as a foundation to discuss subsequent challenges and requirements. The example is presented using the Turtle syntax and is concerned with the description of sensed parking availability at a parking facility.

Listing 2.1 shows the prefix header of the considered document. Prefixes allow us to shorten URIs and make terms and documents as a whole easier to read. The example uses the previously introduced W3C recommendations (RDF, RDFS, OWL) and further well-known ontologies, i.e., the schema.org vocabulary [81], the Sensor, Observation, Sample, and Actuator (SOSA) and SSN ontologies [109], and, lastly, the Open Mobility Vocabulary (MobiVoc) vocabulary that describes mobility concepts. An overview of all ontologies referenced in this dissertation is provided in Appendix B.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix schema: <http://schema.org/> .
@prefix sosa: <http://www.w3.org/ns/sosa#> .
@prefix mobivoc: <http://schema.mobivoc.org/#> .
```

Listing 2.1: Prefix header of the turtle file.

An excerpt from the `mobivoc` ontology that describes the parking facility concept is presented Listing 2.2. It presents the definition of two classes (`mobivoc:ParkingFacility` and `mobivoc:ParkingFacilityEntrance`) and shows how the meaning of the concepts can be described in different languages through `rdfs:label` and `rdfs:comment` properties. Other properties defined as part of RDF, RDFS and OWL further allow to specify relations and constraints of concepts, such as taxonomies by defining relations with `rdfs:subClassOf`, as shown in Line 5 and 18. OWL provides more expressive logical constraints of concepts, such as `owl:equivalentClass`, meaning two classes, potentially defined in different ontologies, describe the same real-world concept (Line 6). Lastly, the example includes the specification of a custom property (`mobivoc:entrance`), stating that instances of class `mobivoc:ParkingFacility` can have an entrance that is defined by the class `mobivoc:ParkingFacilityEntrance`.

```
mobivoc:ParkingFacility rdf:type rdfs:Class ;
   rdfs:label "Parking facility"@en ;
   rdfs:label "Parkeinrichtung"@de ;
   rdfs:comment "Any facility or area assigned for parking vehicles. [...]"@en ;
   rdfs:subClassOf schema:CivicStructure ;
   owl:equivalentClass schema:ParkingFacility .

mobivoc:rateOfOccupancy rdf:type rdf:Property ;
   rdfs:label "rate of occupancy"@en ;
   rdfs:comment "Indicates the percentage value of parking spaces occupied in a parking
       facility."@en ;
   rdfs:domain mobivoc:ParkingFacility ;
   rdfs:range xsd:string .

mobivoc:ParkingFacilityEntrance rdf:type rdfs:Class ;
   rdfs:label "Entrance"@en ;
   rdfs:label "Einfahrt"@de ;
   rdfs:comment "Entrance of a parking facility where vehicles can enter the parking
       facility."@en ;
   rdfs:subClassOf schema:CivicStructure .

mobivoc:entrance rdf:type rdf:Property ;
   rdfs:label "hat Einfahrt"@de ;
   rdfs:label "entrance"@en ;
   rdfs:comment "Describes the entrance of a parking facility."@en ;
   rdfs:domain mobivoc:ParkingFacility ;
   rdfs:range mobivoc:ParkingFacilityEntrance .
```

Listing 2.2: Excerpt of an ontology defining the parking facility concept.

Listing 2.3 demonstrates how concepts defined in an ontology can be instantiated to describe data. The excerpt shows one parking facility in the city of Lyon with two entrances, one for pedestrians and one for cars.

```
_:LPA0764 rdf:type mobivoc:ParkingFacility, sosa:FeatureOfInterest, sosa:Platform ;
   rdfs:label "Parking LPA Hotel de Ville"@fr ;
   schema:openingHours "Mo-Su" ;
   mobivoc:entrance _:LPA0764-101 ;
   mobivoc:entrance _:LPA0764-201 ;
   mobivoc:capacity [
      mobivoc:validForVehicle mobivoc:Car ;
      mobivoc:maximumValue 200 ;
   ] .

_:LPA0764-101 rdf:type mobivoc:ParkingFacilityEntrance ;
   mobivoc:pedestrianAccess "false"^^xsd:boolean ;
   schema:geo [
      schema:latitude "45.768522" ;
      schema:longitude "4.83773" ;
   ] .

_:LPA0764-201 rdf:type mobivoc:ParkingFacilityEntrance ;
   mobivoc:pedestrianAccess "true"^^xsd:boolean ;
   schema:geo [
      schema:latitude "45.768241" ;
      schema:longitude "4.836341" ;
   ] .
```

Listing 2.3: A linked dataset describing a parking facility instance based on the concepts defined in the ontology.

The aforementioned example describes rather static or rarely changing aspects of the parking facility. This characteristic of data is common in datasets from the Linked Open Data (LOD) cloud. In IoT settings, however, new data is continuously sensed and published as data streams. The example in Listing 2.4 illustrates how a sensor observation about the percentage of currently available parking spots at the previously introduced parking facility is represented using the SOSA ontology. The observation states that 90% of the parking spaces of the parking facility are occupied.

```
_:LPA0764_availablility_1575361113 rdf:type sosa:Observation ;
    rdfs:label "Parking availability at parking facility LPA0764"@en ;
    sosa:hasFeatureOfInterest _:LPA0764 ;
    sosa:hasSimpleResult "90" ;
    sosa:resultTime "2019-12-03T08:18:33Z"^^xsd:dateTime ;
    sosa:observedProperty mobivoc:rateOfOccupancy .
```

Listing 2.4: A sensor reading describing information about available parking spaces.

This simple example highlights some of the advantages of using Semantic Web technologies. Ontologies and vocabularies allow for rich declarations of concepts and their relations. Logical constraints further allow to match and link resources of different ontologies of similar and related domains. The descriptions are human- as well as machine-readable, and since best-practices include making the definitions of the concepts available at their URI, the documentation can be easily accessed on the Web.

To present the example more comprehensively, we visualize the corresponding RDF graph in Figure 2.1. The graph only includes a few selected concepts for demonstration purposes, while the actual specifications are much larger. The figure illustrates how Semantic Web standards, ontologies/vocabularies (i.e., classes, relations, constraints, etc.) and data (i.e., instances of classes and properties, including metadata) are represented with the same formalism so that the model- and instance-level are not clearly separated. Vocabularies themselves are expressed as *Web Data*, and thus, Semantic Web tools often do not clearly distinguish these levels. Practically, it can be challenging to make a clear distinction between these levels: even in the scope of ontologies, some concepts may be modeled as instances (e.g., `mobivoc:Car`), highlighting the difficulty in making a distinction of schema and data. The example will be used in the following section to demonstrate the challenges for ontology reuse when engineering ontologies and modeling data.

2.1.3 Ontology Reuse and Recommendation

It is a fundamental best practice in the Semantic Web to reuse existing ontologies and to establish semantic relations between terms [26]. The key advantage and success factor of ontologies as knowledge representation are its characteristics of being unambiguous, adaptable, shareable, and reusable [116]. However, in practice, ontology reuse can be a cumbersome task, and the capability of efficiently finding and selecting most relevant ontologies and terms that already exist on the Web is crucial to ultimately foster interoperabilty [204]. In this context, this dissertation is concerned with *ontology recommendation*, i.e., the task of recommending the most suitable ontology or term to a user for reuse, with a particular focus on ontology reuse in the IoT. We refer to tools that support this task with the umbrella term *Ontology Recommendation Tool (ORT)*.

Figure 2.2 provides a conceptual overview of ontology recommendation and its role in the scope of ontology reuse. We consider four aspects: ontology creation and publication, discovery, selection, and integration, for which we provide greater insights in the following.

Creation and publication. Ontology engineering is a typical task that involves the

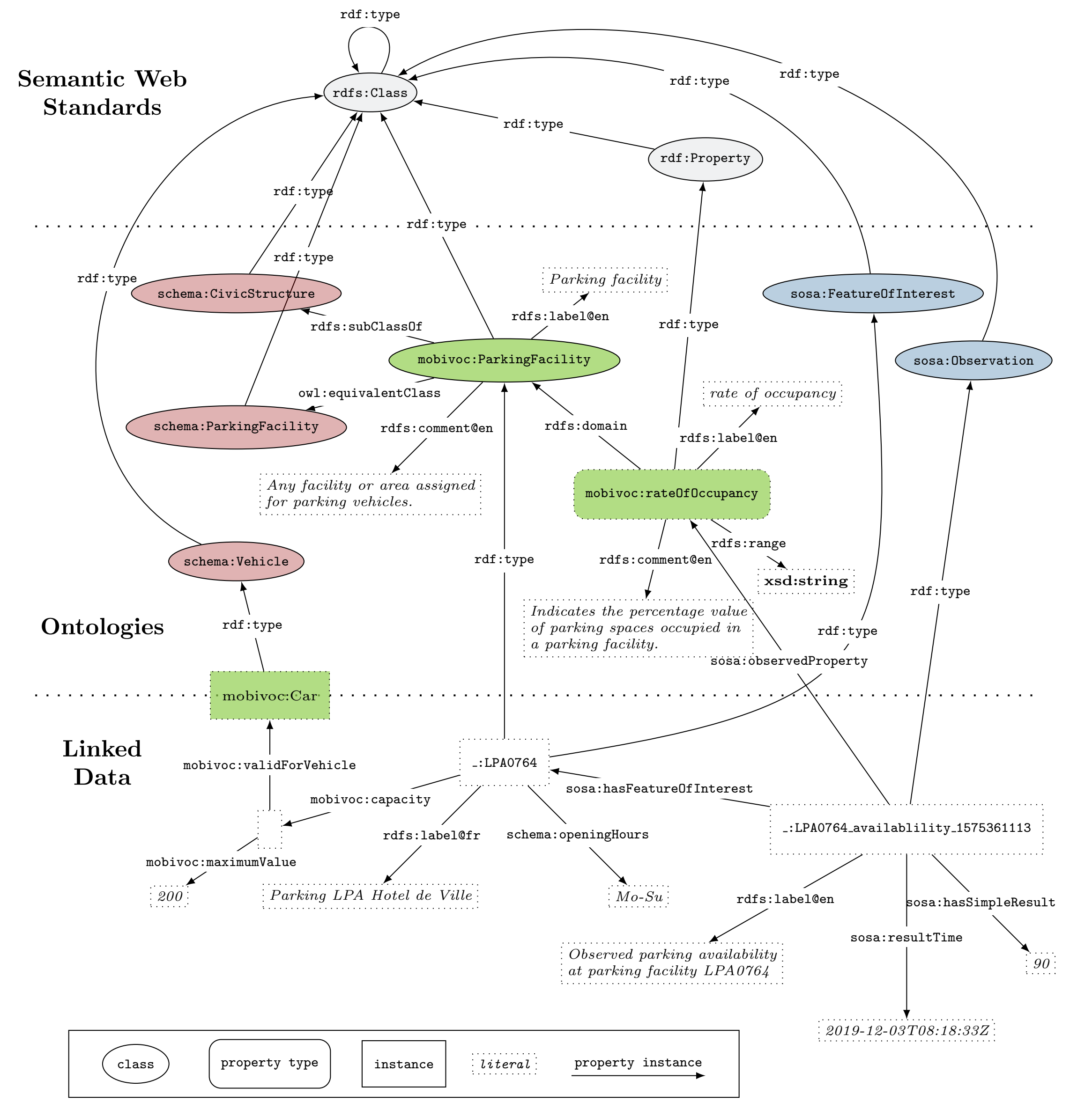


Figure 2.1: Example RDF graph visualized.

challenge of ontology reuse and recommendation. When developing new ontologies, such as the MobiVoc ontology (cf. Figure 2.1), it is vital to establish linkage to existing concepts [77]. From the perspective of the MobiVoc engineers, it is crucial to be aware of already existing ontologies and concepts in the mobility domain, e.g., to decide to make mappings to `schema:ParkingFacility` and not to define a new term for vehicles, but rather reusing the term `schema:Vehicle`. Similarly, terms and ontologies may exist that contain similar concepts but are not important or relevant for the purposes and scope of MobiVoc. Moreover, mappings and linkages among ontologies need to be maintained throughout their whole life cycle. However, ontology engineering and its sub-challenges are out of the scope of this dissertation. More details about ontology engineering can be found in the literature [214, 135, 77].

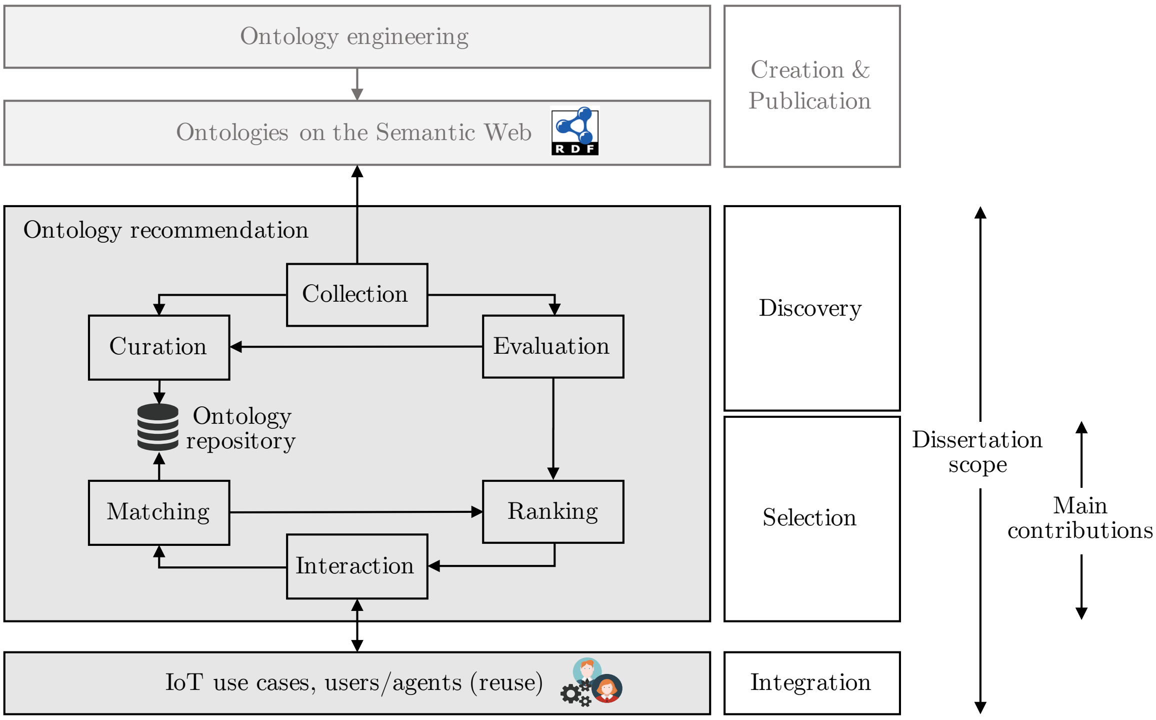


Figure 2.2: Ontology recommendation overview.

Discovery. It forms the first step of ontology recommendation, given existing ontologies on the Web. In order to be able to recommend ontologies, it is required to build and maintain an ontology collection as a repository of ontology or term candidates that may be recommended for reuse. This discovery process comprises the *collection*, *evaluation*, and *curation* of ontologies, which are respectively concerned with finding/gathering existing ontologies on the Web, assessing their quality, and maintaining the repository of suitable candidates. Discovery and assessment of ontology candidates is for example studied in the context of Web crawling [64] and ontology catalogs [60].

Selection. It forms the second step of ontology recommendation. Given a collection of ontology candidates, ORTs further need to provide efficient search mechanisms to users. The selection process is composed of *interaction*, *query matching*, and *ranking*, which are respectively concerned with providing intuitive interfaces for users/agents, finding a match of suitable candidates in the repository based on a query, and ranking these candidates for ontology/term recommendation purposes. The selection of ontologies for reuse is considered in the literature for ontology search and ranking [38] as well as reuse [115]. The main contributions of this dissertation relate to selection, which, to a large extend, is an Information Retrieval (IR) problem. Thus, we present detailed background Information Retrieval (IR) techniques relevant in the context of ontology selection in Section 2.2.

Integration. The design and implementation of discovery and selection features of ORTs are driven by the use cases that they should support. E.g., an ORT may be designed for a specific domain and only collects ontologies relevant to it, some tools may be designed to find best-covering ontologies to annotate a text, while others aim to find a fitting term for a keyword. ORTs are often designed to support a popular use case from the Semantic Web perspective, namely the ontology engineering task (cf. Figure 2.2). However, this thesis is concerned with novel use cases that emerged through the IoT ecosystem vision. In the following, we provide two relevant perspectives.

Data annotation. When adding semantic annotations to data, engineers need to choose terms from at least one suitable ontology. In an open Web setting, we assume that there is no industry- or company-wide standard imposed on the engineer. In case of the sensor reading publication from the example in Figure 2.1 (white nodes), engineers need to choose one or more ontologies to represent sensor(s) and sensor observations in general, as well as observed objects, such as parking facilities and their parking availability based on SOSA and MobiVoc as in this example. Only in rare cases, a single ontology will satisfy all needs. Often, the data itself only establishes the relation between ontologies: SOSA and MobiVoc are completely distinct ontologies, but the data connects these two, e.g., by defining `mobivoc:rateOfOccupancy` being the `sosa:observedProperty` of the sensor reading. Since ontologies and data schemas can be very complex, and many ontologies for the same domain may exist in parallel, finding and choosing the best combination of ontologies is a challenging task during data annotation. Respective IoT use cases include the provisioning of smart city portals [15] and, in general, exposing IoT data to semantic Web gateways and linking them to existing knowledge on the Web [174].

Data discovery and integration. Lastly, relying on existing ontologies and terms can help to find datasets by avoiding ambiguity that is, e.g., imposed by a keyword-based search, and it allows discovering data in an expected format that provides eased (or even automated) integration in the subsequent data processing. E.g., specifying search by looking for instances of `mobivoc:ParkingFacility` (and semantically equal concepts) guarantees that resulting datasets correspond to the given format, while a broader keyword-based search may result in datasets that cannot be processed by the system. In the context of the IoT, e.g., this concerns the development of context-aware applications [200, 172] and respective search engines [71].

In Chapter 4, we provide a novel conceptualization of ORTs along with a detailed evaluation of the state-of-the-art. We further extend the discussion of ontology recommendation by identifying its standing and limitations for the integration in today's IoT use cases.

2.2 Information Retrieval

Information Retrieval (IR) can be defined as *"finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)"* [139]. One aspect of ontology recommendation, i.e., selection, is partially concerned with *search*, which is defined as an IR task. In this section, we present the background for relevant IR techniques that are employed in the context of ontology search, including relevance and importance scoring, Learning To Rank (LTR), as well as the evaluation of ranking models.

2.2.1 Relevance and Importance

The simplest case for IR, one that does not allow to rank documents, is the boolean retrieval: documents either match or they do not, e.g., by selecting all documents in the corpus that contain all keywords of the query. However, in many cases, this results in inconvenient results (especially when many documents are matched) as users need to go manually through the complete set of matched documents. Given that the ranking algorithm works efficiently, a large number of matches in *ranked* IR are not an issue, as the most relevant results are shown on top, and the information need is ideally satisfied by checking only one or a few documents.

Fundamentally, ranking models assign a score to all documents in the corpus for a given query, also referred to as query-document pairs. A ranking model outputs matched documents by the descending order of their scores so that documents with higher scores appear on top. In this thesis, we refer to such scoring functions as *ranking models*, which can be categorized into *relevance* and *importance* ranking models [132]. Relevance ranking models rank documents based on their relevance to the query, while importance models compute scores for documents based on their standing in the corpus (typically independent from the query).

The intuition behind relevance ranking models is about scoring how close the query words are to the words in the documents and how frequently they appear in the documents – the more often a word from the query appears in the document, the higher should be the assigned score for that document. One of the most well-known and reliable scoring functions for this purpose is the Term Frequency–Inverse Document Frequency (TF–IDF) measure [194]. Documents are ranked higher with a higher frequency of the query term in the documents (Term Frequency (TF)) and increased rarity of that term in the entire corpus (Inverse Document Frequency (IDF)). Several related models have been proposed that build upon this intuition and have also been proven to score relevance efficiently in practice, such as the BM25 ranking model [188].

Apart from relevance, one may be interested to take certain characteristics of a document into account to determine its overall rank, i.e., importance ranking models. The characteristics that could be useful criteria to be considered for ranking highly depend on the document type. In Web search, e.g., an importance ranking model may take into account how often a Web page was referred to from other Web pages, such as by the well-known PageRank algorithm [164]. In the simplest case, in order to take the query into account, importance features can be combined with a boolean retrieval, i.e., only matching documents are ranked based on their importance score. However, in general, it is reasonable to take several criteria into account when developing a ranking model, and combining various scoring functions is not trivial, as discussed in the following section.

2.2.2 Learning To Rank

Several relevance and importance scoring functions are often combined to build a better performing ranking model, however, manually combined scoring functions require an expert's intuition about the weight each function should contribute to the final ranking score [132]. This is a challenging, error-prone, and costly task, since it may require a lot of iteration and evaluation steps in order to achieve a satisfying performance. Machine learning techniques can help to overcome this issue. A class of supervised machine learning algorithms called Learning To Rank (LTR) allow us to automatically tune the parameters when combining different ranking models [132]. In a LTR context, we refer to scoring functions as *features* and query-document pairs are represented by *feature vectors*.

LTR algorithms can be classified into pointwise, pairwise and listwise approaches that differ in terms of training input and predicted output, used hypotheses and loss functions [132]. E.g., the pointwise approach takes as input feature vectors for single documents and predicts the relevance score for each document; the pairwise approach relies on pairs of documents (represented by feature vectors) and predicts the pairwise preference; and listwise algorithms take as input a set of documents and predict a ranked list [132]. Input formats can be converted to meet the requirements of each of the three approaches.

A typical LTR workflow consists of the following steps (cf. [134]).

- Following some strategy to sample some *query-document pairs* (it is usually not feasible to define all possible queries, nor to judge all documents in the corpus for each considered query);
- for each query-document pair in the sample, collect target labels that define how relevant the document for the query is (often expressed through a score from 0-4), forming the *ground truth*;
- define useful scoring functions for the problem at hand and, for each query-document pair in the sample, extract the scores (resulting in *training*, *validation*, *test sets*);
- learn and evaluate the ranking model with a LTR algorithm in a classical machine learning manner (hyperparameter tuning, evaluation metric selection, cross-validation, etc.);
- deploy the ranking model, usually realized by *re-ranking of top-k documents* of a simple search result since complex ranking models would not be performant on the whole corpus or large results sets from previous boolean retrieval.

One of the major challenges in this workflow is to evaluate ranking models: it is difficult to assess whether a ranking model succeeds in satisfying users' information needs. In the following, we discuss ranking evaluation in more detail.

2.2.3 Ranking Evaluation

Evaluation of developed ranking models is an important task to assess their effectiveness in helping users to satisfy their information need. In the LTR workflow, ranking evaluation is concerned with defining the target relevance labels of the ground truth as well as the formulation of metrics that measure how well ranking models perform on the ground truth, as detailed in the following.

Building a representative ground truth can be achieved by relying on human annotators or assessors [139] that provide their judgment on how relevant a document for a certain query is (either in a pointwise, pairwise or listwise form) [132]. However, manual judgments are very time consuming and thus costly, may require expert as well as domain knowledge, and could be biased if the number of assessors is low [236]. This is particularly a problem for ranking models built through LTR, as these require large datasets for training, validation and testing. Thus, particularly in the domain of Web search, so-called ground truth mining approaches have been proposed to create large ground truths more cost-efficiently. A common approach relies on *implicit* user feedback that is collected by observing query and clicks made by users. This input is used by user click models [48] to infer actual relevance labels from the noisy click logs. Whichever approach is followed to create a ground truth, its correctness is fundamental to ensure that a ranking model's evaluation is based on a representative foundation. Mining approaches usually also come with the disadvantage that they can only evaluate ranking models' performance based on the user's query, while the actual intent and information need are not known.

The evaluation of IR methods, given a ground truth, rely on *precision* (fraction of relevant documents from all retrieved documents) and *recall* (fraction of relevant documents in corpus that were retrieved), however, the importance of these two metrics depends on the case. For Web search, typically a high precision is more important: the main criteria is that the most relevant documents are ranked on top, as the user may be able to satisfy his information need with one or a few documents, and has no interest in finding and looking at *all* relevant documents. I.e., the circumstance that further relevant documents that are not identified as relevant by the model may exist in the corpus is often only a secondary concern. Following this intuition, several popular evaluation metrics are commonly used in the context of Web search to assess a ranking model's performance based on a given ground truth. We limit our discussion to the metrics used in this thesis: Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG) and Expected Reciprocal Rank (ERR).

Mean Average Precision (MAP) [13] is an evaluation metric that computes a single measure between 0 and 1 for a set of queries (e.g., test queries from a ground truth). MAP computes the average (arithmetic mean) of the Average Precision (AP) for each query, and the AP determines the average precision value for a set of top k documents for a single query. The precision at k (P@k) is defined by the fraction of relevant documents in the top k results returned by the ranking model:

$$\mathrm{P@}k = \frac{\sum_{t \leq k} \mathrm{rel}(k)}{k} \tag{2.1}$$

where rel(k) equals to 1 if the document at position k is truly relevant and 0 otherwise. Subsequently, the average precision for a query Q is defined as:

$$\mathrm{AP}(Q) = \frac{\sum_{k=1}^{m} \mathrm{P@}k * \mathrm{rel}(k)}{|\mathrm{rel}_Q|} \tag{2.2}$$

where $|\mathrm{rel}_Q|$ corresponds to the number of relevant documents for query Q, and m being the number of associated documents for query Q (i.e., number of documents in the ground truth with a relevance judgment for query Q). Finally, the overall MAP score for a ranking model is computed by:

$$\mathrm{MAP} = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} AP(Q) \tag{2.3}$$

where $\mathcal{Q}$ corresponds to the set of test queries that are considered in the evaluation. MAP has the following characteristics worth mentioning. First, each query factors equally into the final score, even though some queries might have very few and others a lot of relevant documents associated with it, often leading to a high variation of AP scores among queries [139]. Second, MAP, as considered here, is dependent on precision at k – a practical necessity since it is hard or impossible to obtain an estimation of how many documents are relevant for a query in the entire corpus. When calculating AP, the cut-off value k may not be fixed for all queries: it is determined by the number of judged documents for the query (parameter m). Since the considered k has a significant impact on the P@k score, it is considered as one of the *"least stable of the commonly used evaluation measures"* [139]. From a LTR perspective, this observation implies that the design of the ground truth impacts the observed performance of the ranking model. These effects are studied in the LTR community under *sample selection* [150] as well as *deep/shallow judgments* [245] and are important criteria when interpreting ranking model performance results. Third, MAP only considers binary relevance: it only considers relevance determined by rel(k) in terms of 0 and 1; it does not distinguish between documents that are more relevant than others, e.g., on a discrete scale from 1 to 4. In contrast, the metrics presented in the following do consider such multi-valued relevance labels. Nonetheless, MAP is widely used and, in general, has shown stability for the evaluation of ranked IR models [139].

The evaluation measure **Normalized Discounted Cumulative Gain (NDCG)** [111] considers graded relevance scales and further a position discount factor, meaning that more relevant documents receive higher scores and are further penalized when they appear at later positions in the rankings. Like P@k, NDCG is computed for fixed top k documents of the result list. NDCG computes the Discounted Cumulative Gain (DCG) for each query, which is composed of the (cumulative) *gain* G and the *discount factor* η:

$$\mathrm{DCG@}k(Q) = \sum_{j=1}^{k} G(j) * \eta(j) = \sum_{j=1}^{k} \frac{2^{\mathrm{rel}(j)} - 1}{\log_2(j+1)} \tag{2.4}$$

where rel(j) corresponds to the relevance score (typically between 0 and 4) of the document ranked at position j, $2^{\mathrm{rel}(j)} - 1$ being a typical way of computing the gain G and $\frac{1}{\log_2(j+1)}$ being a common implementation of the discount factor η. NDCG further normalizes the DCG score for each query by its ideal value, i.e., the DCG score of a perfect ranking:

$$\mathrm{NDCG@}k = \frac{\sum_{Q \in \mathcal{Q}} \mathrm{DCG@}k(Q)}{\sum_{Q \in \mathcal{Q}} \mathrm{idealDCG@}k(Q)} \tag{2.5}$$

where idealDCG@$k(Q)$ is the DCG score computed based on the ideal order of the judged documents for query Q. Like MAP, NDCG forms a single measure for a ranking model's performance with a value between 0 and 1. Similarly, NDCG's intuition lies in measuring precision, not recall. The approach to designing the ground truth also impacts the observed NDCG scores, mainly due to the chosen relevance scale and the relevance label distribution for judged documents [150]. NDCG is one of the most popular metrics in Web search settings with graded relevance [139].

The proposed **Expected Reciprocal Rank (ERR)** [43] metric, which also supports graded relevance scales, is designed with the consideration of user behavior in Web search settings. Instead of relying on a simple, position-based discount factor as in NDCG, ERR follows a cascade model that also considers the relevance of higher-ranked documents. Following the intuition of the cascading browsing behavior [54], it takes into account the probability that the user is not satisfied with the first $j-1$ documents but satisfied with the document at position j. ERR relies on a mapping function from the considered relevance grades $\mathcal{G}$ (e.g., $\mathcal{G} = \{0, 1, 2, 3, 4\}$) to the *probability* of relevance $\mathcal{R}$:

$$\mathcal{R}(g) = \frac{2^g - 1}{\max(\mathcal{G})} \tag{2.6}$$

where g is a relevance grade from $\mathcal{G}$ ($g \in \mathcal{G}$). ERR is then defined as:

$$\text{ERR@}k = \frac{1}{|\mathcal{Q}|} \sum_{Q \in \mathcal{Q}} \sum_{j=1}^{k} \frac{1}{k} \prod_{i=1}^{j-1} (1 - \mathcal{R}(\text{rel}(i)))\, \mathcal{R}(\text{rel}(j)) \tag{2.7}$$

where rel(i) corresponds to the relevance grade of the document at position i, and k corresponds to the top k documents for the result of query Q considered for evaluation. ERR evaluates ranking models with the intuition that users will stop looking at further documents if they already have seen one or many documents (assuming a cascading browsing behavior) with high relevance in the ground truth. It is argued that ERR is thus a better metric for Web search engines [43]. In the following, we summarize the case of applying IR techniques in the context of ontology search.

2.3 Ontology Ranking

Ontology ranking and ontology search describe applications of IR techniques to ontology collections. In this section, we present the details and particularities of this task by applying general IR concepts, as introduced in the previous section, to the case of ontologies.

Query. The query format remains the same for ontology ranking: keywords. Less dominant query formats in the scope of ontology ranking include structured query languages and free text.

Documents. Two different retrieval tasks are (unfortunately) both referred to as *ontology* ranking in the literature: documents to be retrieved can either be whole ontologies or individual terms from a set of ontologies. Whether users are looking for ontologies or specific terms depends on their use case and in the case of term ranking, overall criteria of the terms' ontology are similarly important [196]. Similar to Web search, ontologies and terms are not just plain text but rather semi-structured documents, which changes the nature of developing relevance and importance scoring functions.

Relevance. It is essential to determine the relevance of an ontology or term to a query. In ontology ranking, the URIs might be considered as an indicator, however, in practice, the URI does not need to carry any meaning. While it is a fairly common practice to include the concept's name in its URI, Wikidata[4], e.g., creates its URIs based on generated numeric identifiers that do not indicate anything about the meaning about the concept that this URI represents. Thus, relevance search of ontologies needs to consider specific properties of ontologies/terms and apply query matching scores based on strings defined in these properties. The most common ones are `rdfs:label` and `rdfs:comment`. Approaches that rank ontologies as a whole may look into the properties of all its terms instead of determining relevance only based on annotation properties of the ontology.

[4] `https://www.wikidata.org/`

Importance. The task of determining the importance of ontologies has similarities with conventional Web search, meaning respective importance ranking models for Web documents can be adapted for ontology ranking. These ranking models are complemented with novel criteria that are specifically designed for the ranking of ontologies and terms. For example, being referred to by other ontologies gives an ontology a better standing, as it means that it has been used before by others (following the intuition of PageRank from conventional Web search). On the other hand, criteria such as logical consistency, ontology status, availability, etc., may also form important factors when recommending ontologies for reuse that are specific to the task of ranking ontologies. Moreover, specific criteria depending on ontologies' domain and reuse intent can also be of interest.

Evaluation. Ontology ranking adopts the common assumptions from conventional Web search for evaluation, meaning it is assumed that users do not want to go through all matching ontologies/terms, but rather the most relevant ones until their need is satisfied. Thus, precision-based evaluation metrics are applied for ontology ranking.

More details on the state-of-the-art and novel insights on ontology ranking are developed throughout the remainder of this dissertation, in particular in Chapter 3 to Chapter 6.

2.4 Summary

This chapter presents relevant background material for this thesis by introducing techniques in the scope of the Semantic Web, IoT use cases and IR. The following chapter presents state-of-the-art from the literature regarding the application of such techniques to enable semantic interoperability through ontology reuse.

Chapter 3

Related Work

This chapter presents related work with regard to ontology recommendation. We present the proposed tools that fall in the scope of ontology recommendation, studies that specifically focus on the design of ontology ranking models, and, finally, the state-of-the-art in the evaluation of ontology ranking models' ranking quality.

Contents

3.1 Introduction

Several ontology repositories, libraries, search engines and other tools have emerged in order to discover, search and promote the reuse of already defined ontologies and terms. Furthermore, studies specifically dedicated to ontology ranking and their evaluation have been presented in the literature. In the following, we introduce these studies. Similar reviews can be found in the literature [60, 193, 95, 235, 36], which guided the selection of relevant studies.

3.2 Ontology Recommendation Tools

The ORTs presented in the following are ordered chronologically based on the date of their first publication.

Ontokhoj [167] is one of the earliest ontology search engines described in the literature. The project includes an experimental crawling process to find ontologies on the Web using RDF crawler[1]. The query interface is not a direct keyword-based search, but further allows users to select the exact meaning of their search words interactively based on the WordNet[2] dictionary [148]. The crawled ontologies are automatically classified into domains (e.g., university, sports, computer science) and the experiments include a comparison of several classification algorithms (e.g., Naive Bayes, k-nearest neighbors). The proposed ranking mechanism is PageRank-based, with the modification of giving a different priority (weight) to the type of relationship, i.e., hyperlinks, differentiating between three groups: `rdf:type` (highest priority), `rdfs:subclass`, and `rdfs:domain` (lowest priority). The importance-based (i.e., query-independent) ranking model is then applied per domain, scoring respective ontologies. Ontokhoj does not include a formal evaluation of the ranking effectiveness, nor is the prototype available online at the time of writing.

OntoSelect [31] was proposed to provide a tool for ontology selection with a continuous crawling procedure that monitors the Web for newly published ontologies. Thus, it is also built upon a diverse ontology collection of various domains. Newly found ontologies are analyzed in terms of used knowledge representation formalisms, used languages, and size. The proposed ranking mechanism is designed under the aspect of helping users selecting the most useful ontology. Thus, the ranking is composed of three aspects: (i) coverage, a more thorough relevance match adjusted to the structure of ontologies, i.e., taking into account the number of matched labels in the ontology; (ii) structure, a straight-forward importance score that favors ontologies with a higher properties-per-class ratio; and (iii) connectedness, a PageRank-based scoring function. However, no formal evaluation of the proposed ranking model is presented and the respective application is not available online at the time of writing.

Swoogle [64, 65] is an ontology search engine that had a significant impact and was actively used. Swoogle is not only designed to enable search for ontologies but also instances, i.e., linked data, however, it follows a document-centric approach. The underlying collection of Semantic Web documents is built through continuous Web crawling and, over time, led to a collection that includes over 10000 ontologies. The initial set of potential URLs of Semantic Web documents is collected through several Google searches, and users are able to submit URLs for focused crawling. Similarly to OntoSelect, the goal of Swoogle includes analyzing characteristics of the Semantic Web,

[1] `http://ontobroker.semanticweb.org/rdfcrawl/`
[2] `https://wordnet.princeton.edu/`

such as used knowledge formalisms. Swoogle's ranking for keyword-based search is also PageRank-based, however, the adapted scoring function considers a more detailed categorization of relation types among ontologies. Various `rdfs` and `owl` terms are grouped into six relation types: term references, imports, extension, prior version, compatible prior version, and incompatible prior version. The relation type then defines a weight of the PageRank scoring function. The ranking evaluation includes a comparison with an unweighted PageRank approach with the goal of ranking documents that define ontologies higher than others but does not include a formal evaluation on the ranking performance. The platform is still accessible online[3], yet not always fully functional.

Supekar et al. [217] aim to analyze the quality of existing ontologies and provide respective interfaces in order to support the ontology engineering process. This work does not concern the collection of ontologies and relies on the Ontokhoj tool to perform analysis on a real-world ontology collection. The authors argue that a qualitative analysis of ontologies is an important aspects when searching ontologies for reuse and that *"determining the right ontology is still an open issue"* [217]. The proposed features encompass measures about maintainability, usability, availability, re-useability, reputations, citations, semantic expressibility and syntactic correctness. Moreover, the approach proposes a weighted sum for the combination of these features to determine a preference score of each ontology for reuse. However, the qualitative evaluation is performed on an ontology collection of one domain and does not consider these criteria combined with user queries or search features, nor does it provide an evaluation of the proposed criteria or a prototype that is available online.

OntoMetric [133] proposes a method to compare and eventually choose ontologies for reuse. The proposed criteria include the dimensions content, language, methodology, tool and costs, for which detailed lists of ontology characteristics are defined. The proposed method to choose an ontology is based on the Analytical Hierarchy Process (AHP) to select an ontology following these criteria. The tool aims at a manual evaluation of several ontology candidates to assess their suitability for reuse, rather than providing efficient recommendation based on minimal user input (as it is common in ontology search). The proposed methodology is difficult to automate and not applicable to search, since many specific ontology features rely on subjective judgments for quality aspects that need to be made by humans. Nevertheless, this perspective of ontology selection highlights the limitations of search engines and that the results in terms of a ranked list of ontologies and terms need to be reviewed by the user to make a final decision – not all criteria can be automatically extracted and considered upfront in the retrieval system. At the time of writing, the prototype is not available online.

Ontology Auditor [34] suggests several metrics to assess the quality of ontologies. The metrics relate to syntactic, semantic, pragmatic and social quality of the ontologies, defining in total 15 attributes. In contrast to OntoMetric, these attributes are defined in the form of formulas that can be objectively extracted from the ontologies. The Ontology Auditor tool is composed of a crawling module that collects ontologies and a rating module that assigns scores based on the suggested quality attributes. Finally, a total quality score is computed through a fixed weight combination as proposed by the authors, which, however, is not formally evaluated. The assessment results can then be retrieved via a UI and an API, i.e., it does not offer any search capabilities. As of the time of writing, the tool is not available.

OntoQA [223, 222] forms another approach to analyze the quality of ontologies.

[3] `http://swoogle.umbc.edu/2006//index.php`

The metrics address both the schema level (e.g., relationship richness, attribute richness, inheritance richness) and instance level (e.g., class richness, average population, cohesion, importance, fullness, connectivity). The metrics are formally defined and can be directly extracted from the ontologies. The tool does not provide any search features, instead, a user is expected to provide an ontology subject to the quality assessment and the tool computes the quality scores for that ontology. No combination of metric's scores is provided, meaning it requires further individual interpretation to reach an understanding of whether the ontology satisfies the user's information need. Moreover, the proposed metrics are not formally evaluated. While such a tool does not help in finding ontologies, it can support users in analyzing a known set of ontology candidates considered for reuse. To the best of our knowledge, no prototype for this tool is available.

Ontosearch [248, 112] is a search engine that aims to provide a specific user interface and visualizations for ontologies on top of the Google search engines: users' keyword input is forwarded to the Google Web API (which does not exist anymore), filters the result set to ontologies, and the resulting ontologies can be explored visually by the user. This tool employs the fact that ontologies are de facto Web pages and traditional Web search could be applied for ontology search. However, using Google for this purpose has not become a common practice. Ontologies are more complex compared to hypertext documents, and efficient ranking demands adjusted scoring functions for this purpose.

Knowledge Zone and TS-ORS [131, 218] form an ontology repository for biomedical ontologies. The approach focuses on a quality indicator based on user ratings, which was designed to simplify the ontology reuse process. The ontology collection is built through user submissions. The final ranking score of ontologies is built upon a model coined as *Web of Trust* - it is not only based on the ratings of the ontology by the users, but also based on the ratings of users made by other users, i.e., measuring to what degree a review of a user can be trusted. Users are able to tune the weights of two specific ranking features, the trust and the distrust rank, depending on which criterion is more important for the user. In ontology reuse, getting user feedback about ontologies is an important aspect that can help the convergence to common ontologies in a community. On the other hand, feedback in terms of user ratings requires an active community. To the best of our knowledge, the Knowledge Zone platform is no longer available online.

Open Metadata Registry [98] is a platform that goes beyond the discovery of ontologies: it further provides means to create and maintain ontologies with the goal to foster reuse and interoperability even when ontologies evolve and new versions arise. However, the evolution of "external" ontologies is considered out of control. The tool also provides an interface to explore ontologies, such as defined classes, provides authorship roles to define who may edit or expand an ontology and tackles various challenges in the scope of ontology versioning. It further provides a straight-forward relevance search feature for ontologies and terms (for which no detailed information is provided). The tool can be accessed online[4] and highlights the collaborative aspect of ontology engineering and reuse.

Ontosearch2 [166, 225] is an ontology search engine that was developed to supersede Ontosearch. In addition to the ontology exploration of Ontosearch, Ontosearch2 supports lightweight DL reasoning in order to find implicit relationships between on-

[4] http://metadataregistry.org/vocabulary/list.html

tologies. It further describes the addition of a SPARQL query interface to the keyword-based one to support users in finding ontologies and terms more efficiently. It is able to process fuzzy queries in which a confidence threshold can be specified that determines whether a term matches or not. The formal evaluation of Ontosearch2 is based on non-functional requirements such as performance, i.e., the effectiveness of the various search interfaces is unclear. To the best of our knowledge, Ontosearch2 is not available online.

Oyster [165] is a distributed repository that was designed to assist researchers in reusing existing ontologies. It focuses on input provided by users: ontologies and their metadata, such as their domain, are submitted by users, so-called peers. The UI offers a search interface in which queries can be formulated in the form of keywords, and further restricted to matches in specific ontology properties. Moreover, the search can be restricted to ontologies provided by specific peers. Due to its distributed nature, a search may result in a lot of duplicates. Oyster employs a semantic similarity analysis to show a merged view to the user. However, no formal evaluation of the tool's efficiency in terms of ontology reuse is presented. The project's source code is available online[5].

(Web)CORE [72, 41] aims at providing a recommendation for the reuse of ontologies based on an informal domain description. The tool's UI allows to specify and refine the domain description, which is considered as the *gold standard* of the user's information need. Ontologies from the repository are then evaluated based on several levels (e.g., lexical, syntactic, semantic) and evaluated based on their similarity to the gold standard. Additionally, collaborative filtering techniques based on user evaluations are employed to integrate human judgment about the ontology reuse candidates for more complex criteria. Lastly, all criteria are combined to measure the similarity between candidate ontologies and the specified information need by the user, and a ranking model is proposed based on this measure. The combination of scores is based on fixed weights and pre-defined by the authors. The ontology recommendation capabilities of this tool are evaluated based on a user study with 18 participants and 30 ontologies, in which the participants were given a task and had to specify their satisfaction with the tool's recommendation. To the best of our knowledge, WebCORE is not available online at the time of writing.

SWSE [93, 104] is a Semantic Web search engine that puts emphasis on scalability through its distributed implementation. SWSE crawls the Web for ontologies as well as linked data using MultiCrawler [94], and indexes them, resulting in more than $250M$ RDF statements. The application provides a UI to browse the discovered concepts and entities and furthermore performs reasoning to infer additional facts about the viewed concepts. The underlying ranking of the keyword-based search is PageRank-based, which has been separately studied under the name ReConRank (see Section 3.3). However, as of the time of writing, the tool is not available online.

Sindice [229, 162] is a RDF document-oriented search engine with a continuous crawling process that updates Sindice's search index. Through its document-oriented approach, it considers both ontologies and data. The architecture is designed to provide an efficient lookup of the index through a keyword-based search. The tool does not aim to extract further metadata or statistics from the discovered documents, however, it applies reasoning over the indexed documents in order to make implicit statements explicit. The ranking of the search feature takes into account a TF–IDF-based query match, an external ranking score of the document's hosting page, as well as whether the

[5] `https://sourceforge.net/projects/oyster2/`

hostname matches the document's base URI. The cited publications do not describe the composition of the final ranking model. At the time of writing, the application is not available online.

Watson [57, 59] forms a Semantic Web search engine with a particular focus on the analysis of ontologies, arguing that a Semantic Web gateway should take into account semantic particularities. Thus, Watson analyses discovered ontologies to extract metadata, classify their content and identify relations among documents. The extracted metadata enhances the result lists and provides additional information to the users. Moreover, it is argued that such metadata can be a crucial factor in terms of ontology ranking to support the ontology selection task, however, no explicit ranking model taking these criteria into account is proposed. The UI further offers the option to explore the resulting ontologies and provides a SPARQL endpoint. At the time of writing, the application is not available online.

OBO Foundry [206] is targeted to biomedical ontology engineers and advocates shared principles for ontology design with the goal of maintaining a set of interoperable ontologies. The principles include open use, collaborative development, non-overlapping, and strictly-scoped content. Unlike the Web crawling approach, the OBO Foundry ontology collection is administered with well-organized oversight, resulting in an ontology repository with higher quality, more trust by the community, and a higher tendency of reuse. This, eventually, enabling systems' interoperability, however, is restricted to the respective domain and community. The OBO Foundry UI provides a tabular view of the maintained ontologies and includes a link to the Ontobee (see below) search interface. The platform is available online[6].

(combi)SQORE [230, 231] is an ontology retrieval framework that relies on so-called semantic queries. The complex query format allows the specification of mandatory and optional conditions, e.g., with regard to string matches and relations, and to specify a weight for each condition. The framework subsequently defines several scoring functions that are aggregated to a similarity score (that defines how well the terms used in the semantic query match the concepts in the ontology) and a query coverage score (that determines how well the defined constraints are covered by the ontology). The underlying ontology collection is gathered through the Watson search engine, however, neither the UI nor the API of SQORE are available online at the time of writing.

Falcons Concept & Entity Search [46, 181] is a search engine for both the search of concepts (i.e., terms) in ontologies as well as entities from linked data. The platform also follows a Web crawling approach to create the underlying document collection, however, unlike previous search engines, it does not only index literal-valued properties (such as `rdfs:label` and `rdfs:comment`) but instead creates a 'description graph' of each concept that enables efficient search with keyword queries that include, e.g., "subclassof" in the query. The UI further provides a more elaborate summary of concepts (e.g., the main properties of a concept that may originate from different documents) and the possibility to browse ontologies. The Falcons Entity Search is one of the earlier approaches that define a *popularity* scoring function in addition to a relevance measure to rank concepts. Popularity is an importance ranking model that is measured based on the number of Semantic Web documents in which a concept is used to describe data. In the Falcons ranking model, popularity is integrated as a factor of the relevance score in the overall ranking model. The evaluation of the platform is

[6] `http://www.obofoundry.org`

based on a questionnaire to assess the usability and satisfaction with 23 participants. The platform is not available online at the time of writing.

VisiNav [92] forms a linked data search engine with a focus on the user interaction model. One goal is to support the user in defining more complex queries based on the combination of atomic query operations. Upon a keyword-based search, users are able to explore the results with several views (e.g., as list and graph and use facets to explore specific properties of the entities and to navigate through the graph until the user's information need is satisfied. The result display follows the notion of 'topical subgraphs', to show the next steps for navigation starting from a focused concept. The underlying data for the prototype is based on LOD dumps as well as a Web crawl, and several indexes are constructed, including statement, path, text, facet and out-link index. The ranking is solely based on relevance, however, the evaluation is restricted to performance aspects and does not consider recommendation capability. The respective prototype is not available online as of the time of writing.

BioPortal [157] is one of the most adopted and featured ontology platform for the biomedical domain. It offers means to add, maintain, browse and visualize ontologies, a search interface for ontologies and terms, an annotator of free text with terms from biomedical ontologies, an ontology recommender, the possibility to define mappings between terms and to provide peer-reviews of ontologies. Thus, BioPortal focuses on the collaborative aspects of ontology engineering and reuse and has formed a rather large community. Moreover, most functionalities are also accessible via its API. While the search features rely on relevance matches with advanced filtering options for users, the recommender functionality provides a more complex ranking model. The details of the recommender are presented in the 'NCBO2.0' paragraph in Section 3.3. The BioPortal can be accessed online[7].

Cupboard [58] is a system that is targeted at ontology engineers and users interested in reusing an ontology. Ontologies are not automatically crawled from the Web but are submitted by users, who are able to further describe and maintain the ontologies through the features of the tool. As a distinction with other tools, users can create their own space to maintain their ontologies rather than feeding a central collection of ontologies. It then relies on external, crawling-based search engines, such as Watson, for respectively added ontologies to be indexed and searchable for other users. It further supports ontology engineers in defining mappings among ontologies within an ontology space. Moreover, it relies on the previously introduced TS-ORS model to assign ratings to ontologies. To the best of our knowledge, the cupboard prototype is not available online.

MMI [191] is an ontology repository that was built to foster marine metadata interoperability. The MMI Web portal allows users to submit and maintain ontologies, provides versioning and collects additional metadata about them. The ontology collection can be browsed and filtered by users based on several criteria, such as ontology owner, status and type. The tool further offers a keyword and a SPARQL search to find terms in the ontology collection. Internally, keyword searches are translated to SPARQL queries that return a simple result list with all triples containing at least one keyword. Similarly to BioPortal, MMI emphasizes the collaborative aspect of ontology engineering, e.g., by showing author and user name in the result list. MMI is accessible online[8].

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

[8] `https://mmisw.org/ont/`

Ontobee [241, 161] provides means for sharing, visualizing, querying, integrating and analyzing ontologies. It serves as the default ontology browser for ontologies of the previously introduced OBO Foundry and, by 2016, contained 180 ontologies with more than four million of terms of the biomedical domain. It offers a keyword-based search for terms and a SPARQL endpoint to query the ontology collection. To the best of our knowledge, the search feature is solely based on relevance and the overall recommendation capabilities are not formally evaluated. Ontobee also includes a feature to extract statistics and to annotate free text with ontological terms. Ontobee is available online[9].

WebOWL [19] is a Semantic Web search engine with a ranking model particularly adjusted to terms. It employs a Web crawler to continuously discover new OWL and update changes in existing ontologies that form the underlying ontology collection with over one million terms. The discovered data is stored in a triple store. WebOWL includes a dedicated ranking algorithm named 'OWLRank'. The ranking mechanism is a PageRank-adaptation on the term level, meaning it takes into account the links between classes and individuals, not just between ontologies as a whole. The query interface allows specifying keywords for specific fields (ontology, characteristics, property restrictions) as well as an OWL-based input in which the statements form the search constraints and a dedicated OWL class specifies the information need. As of the time of writing, the prototype is not available online.

LODstats [12, 69] is a tool that aims to extract statistics from datasets of the LOD cloud. The 32 considered criteria include, e.g., the use of ontologies, classes and properties, as well as quality analysis such as the PageRank measure for ontologies. The underlying document collection is extracted from a LOD catalog. The tool further offers a keyword search based on relevance to find statistics for certain ontologies and terms, which can be accessed via a UI and an API. While LODstats does not primarily aim at recommending ontologies, it extracts valuable metadata that can be used for more enhanced recommendation approaches. At the time of writing, the platform is not accessible online, but the source code is publicly available[10].

vocab.cc [208] provides a search for ontologies used in datasets from the LOD cloud, specifically those from the Billion Triple Challenge (BTC) dataset[11]. The huge datasets are analyzed and the frequency of terms occurring in these datasets is extracted. The tool offers a keyword-based search and matching terms are ranked according to their frequency. The ranking can be accessed via a UI and an API. For users who solely want to rely on how often a term has been used in the BTC datasets, vocab.cc provides easy and efficient access to it and thus potentially recommendations for term reuse. The tool can be accessed online[12].

BiOSS [142] is a system designed for the selection of biomedical ontologies. The tool is based on a collection of 200 pre-selected biomedical ontologies. The user input can be specified in the form of keywords that should describe the domain of interest; furthermore, users can assign a weight to these keywords to define their relative importance, allowing the system to prioritize matches of more important concepts. The tool performs interactive pre-processing on the query, such as spell-checking, avoiding disambiguation and performing query expansion. BiOSS recommends either single or

[9] `http://www.ontobee.org`
[10] `https://github.com/AKSW/LODStats`
[11] `http://km.aifb.kit.edu/projects/btc-2011/`
[12] `http://vocab.cc/`

combinations of ontologies based on four criteria: input coverage, knowledge richness, knowledge formality and popularity. The popularity of an ontology is computed based on the number of papers in PubMed[13] that mention the ontology, whether the ontology is contained in the BioPortal platform, and the number of Wikipedia[14] articles that mention the ontology. The final ranking model combines these criteria with a sum and fixed weights of the criteria. The evaluation includes a questionnaire with 16 participants, which assesses the usability and accuracy of BiOSS. To the best of our knowledge, the respective Web tool is no longer available online.

Manchester OWL Repository [143] is a resource that advocates the curation of ontologies and offers means for creating and sharing ontologies. The ontology collection is composed of crawled ontologies from other repositories such as BioPortal as well as a manually curated set of ontologies. Ontologies must cohere to minimum quality standards, such as syntactic correctness. The repository can be accessed via a UI and an API. The interfaces rely on a query language that allows defining constraints on specific metrics, such as the count of classes or axioms (e.g., 'class_count $\geq X$'). The repository is online available[15], however, as of the time of writing, not with its full functionality.

smartcity.linkeddata.es [175] provides an ontology as well as a dataset collection for smart city use cases categorized into twelve domains that were identified as relevant for such use cases. The collection is built by experts based on IoT project experience, online search and dataset investigation. For each ontology, preset metadata is added, such as ontology title, creation date, syntax, domain, etc. Ontologies are further evaluated based on whether they comply with LOD best practices. The catalog is made available with a UI through which users can suggest ontologies to be added to the collection. However, no recommendation and search features are implemented. The UI solely offers a filter by domain. The catalog is available online[16].

OUSAF [9] is a framework that was designed to analyze the usage of ontologies from RDF datasets. The ontologies are collected from other search engines (i.e., previously introduced Watson and Sindice). The tool extracts various statistics, such as the number of instances, ontology and term usage, interlinkage from the ontology collection. It further computes an importance ranking score for all terms, which is based on richness, usage, and incentive. All metrics are formally defined and can be directly extracted from the dataset. The extracted information, including the ranking, can be retrieved through SPARQL queries. To the best of our knowledge, the prototype of this tool is not available as of the time of writing.

Ontology Lookup Service [114] provides a UI and an API to access biomedical ontologies. It provides a keyword search with optional filters (e.g., term type, specific ontologies) to retrieve concepts defined in the ontologies. Ontologies are not maintained within the platform; instead, external sources containing ontologies are regularly checked for changes and the platform only keeps the most recent version of an ontology in its collection. New links to ontologies can be added through user submissions that are reviewed by the maintainers of the platform. The service can be accessed online[17].

[13] `https://www.ncbi.nlm.nih.gov/pubmed/`
[14] `https://www.wikipedia.org/`
[15] `http://mowlrepo.cs.manchester.ac.uk`
[16] `http://smartcity.linkeddata.es`
[17] `https://www.ebi.ac.uk/ols`

LOV [232] is a curated ontology collection with the aim of facilitating the reuse of common ontologies. The tool provides an ontology, term and SPARQL search/query interfaces that can be accessed through a UI and an API. LOV keeps track of ontologies' versions and extracts metadata of the corpus, such as the relationships and their types among the ontologies, which are further visualized in the UI. Adding and maintaining ontologies is a semi-automated process and ontologies need to meet minimum quality requirements such as dereferencability, syntactic correctness, `rdfs:label` annotations for all terms, etc. The process is overseen by human curators to ensure and reinforce the quality of the corpus. The search feature is keyword-based and the ranking combines a BM25 query match score with a popularity measure extracted from usages in LOD datasets. LOV is a popular platform that has been integrated into many external tools that require ontology or term lookup. The platform is accessible online[18].

Ontohub [50] is a platform that aims to bring ontologies with different knowledge formalism together, i.e., to enable the mapping to terminologies that are not represented in RDF. It relies on the Distributed Ontology, Model and Specification Language (DOL), an umbrella representation for ontologies in several different knowledge representation formalisms. Ontohub is merely composed of a set of repositories and describes itself as repository engine. Furthermore, repositories are Git repositories which allow straight-forward version control supporting the evolution of ontologies. The use of DOL expands the potential reuse of schemas and mappings to other terminologies and standards beyond Semantic Web technologies. The UI offers a keyword search and filtering options over all ontologies/terminologies across repositories. Ontohub, currently in its beta stage, is online available[19].

In this section, we reviewed a selection of ORTs from the literature. In the following section, we present related work that focuses particularly on the ontology ranking aspect.

3.3 Ontology Ranking Models

In addition to the previously introduced tools that aim to collect and eventually recommend ontologies, there are related studies that focus particularly on ontology ranking, a fundamental aspect of ontology recommendation. In the following, we introduce these works, again in chronological order based on the year of publication.

ReConRank [102] is a ranking model designed for the ranking of Semantic Web data, i.e., it is not specifically designed for the ranking of ontologies. It combines a collection-wide PageRank-based importance score (ResourceRank) with a relevance score based on a 'focus subgraph' that depends on the query (ContextRank). The evaluation of this approach focuses on the efficiency in terms of computing time upon a query. Since PageRank-based scores are query-independent, they can be pre-computed. The ContextRank ranking model, however, depends on the subgraph size that depends on the query. The subgraphs are constructed by selecting in- and out-links of the subject node of all matching literals, to which the same scoring algorithm is applied. Unlike standard PageRank adaptations, the latter measures the *authority* of the matched concepts compared to nearby concepts. ReConRank has been used as the ranking model in the previously introduced SWSE search engine [93].

AKTiveRank [6] proposes four scoring functions for the ranking of ontologies.

[18] `https://lov.linkeddata.es/`
[19] `https://ontohub.org/ontologies`

The class match measure is a relevance score based on exact and partial matches in `rdfs:label` properties of ontologies' classes. The density measure is an importance measure based on the number of subclasses, superclasses, relations and siblings defined in the ontology. The semantic similarity measure is an adapted general graph scoring relevance measure based on the shortest path of matched classes in an ontology. Lastly, the betweenness measure adapts a general graph scoring importance measure that measures the centrality of a node based on the number of shortest paths that pass through it. The study further proposed fixed weights for the combination of these features. The evaluation of the ranking model is performed on a dataset that contains human relevance judgments for ~ 400 query-ontology pairs.

LOVR [211] is designed to find ontologies and terms that are most suited to annotate a given Web page. With Web pages forming query input, the framework extracts relevant concepts, which in turn are used to query the LOV platform for relevant terms. Moreover, previously introduced vocab.cc and LODStats are used to extract usage statistics for these terms. The final ranking score combines the LODStats measure, the vocab.cc measure, the LOV ranking score and a boolean match factor (0 or 1) with a fixed formula and is applied to each extracted concept of the Web page. LOVR is practically limited because the employed tools do not have the same underlying ontology collection, i.e., not all considered metadata scores can be extracted for all recommendation candidates. The evaluation only includes a comparison of the framework's recommendations with a single real-world Web page that already contains semantic annotations (thus considered as ground truth). Moreover, the recommendation only considers individual concepts and does not analyze which ontology might be able to cover most concepts of the Web page. Lastly, Web page annotations in practice usually rely on the schema.org ontology, as this is the only one considered by major search engines. Hence, the applicability of Web pages as query input is limited.

DWRank [40, 37] proposes a combination of three scoring functions for the ranking of ontology terms: a boolean match (scoring 1 if a term matches the query, 0 otherwise), a PageRank-based score (determining the authority of a term), and a PageRank-based score based on a reversed ontology graph (determining the centrality of a term). The study also stresses the need to consider implicit relationships between ontologies, as these are often missing, in order to obtain more reliable PageRank-based scores. The ranking model is then built using LTR based on the CBRBench dataset (see Section 3.4). The training data further include two additional features, namely the maximum and minimum centrality score. The study finds that the combination of ontology ranking features as proposed outperforms several classical single-feature retrieval methods adjusted to ontology search. However, this study is limited to the representativeness of CBRBench and the circumstance that it compares a *term* ranking model with *ontology* ranking models, as discusses in Section 3.4.

TermPicker[197] aims to provide a ranking of ontology terms to complete a triple. E.g., given a triple with a missing subject, predicate or object (forming the query), it provides a ranking of terms to fill the gap(s). The underlying data is extracted from datasets of the LOD cloud based on how often a certain combination of terms in triples appears. Therefore, such a ranking model does not support users in finding terms from scratch but helps to complement the user's work with recommendations when already some terms are known. Termpicker also adopts a LTR workflow and experiments with five ranking features; three importance features: number of datasets using the term, number of datasets using the ontology of the term, total number of occurrences of a term; and two relevance features: whether the term belongs to an ontology of a term already used in the query-triple and the observed number of combinations between the term and those provided in the query-triple. The tool's evaluation compares the use of only using importance feature to using additional relevance features – and the experiments confirm that relevance features are necessary for effective ranking. This study simplified the problem of applying LTR to term search: the big challenge is to create a large, representative ground truth to perform empirical experiments. By relying on triple patterns, the need for relevance judgments for query-term pairs could be avoided. However, this limits the application to the ranking model to cases where some terms are already being used, and its main strength lies in assisting the completion of triples based on already seen triples.

RecoOn [39] aims at the recommendation of ontologies based on a text corpus as the query format. The pre-processing includes the extraction of keywords from the query string and translates the keyword query subsequently into a graph and then into a SPARQL query. The SPARQL query is used to retrieve matching ontologies from the repository, which are further considered for evaluation and ranking. The recommendation, i.e., the scoring functions, considers the matching cost (relevance), informativeness (relevance) and 'popularity' (PageRank-based importance). The three ranking features are then used to define a knapsack optimization problem using CBR-Bench. Even though CBRBench only contains query-*term* pairs for single-keyword queries, the ground truth is modified to generate query-*ontology* pairs for multi-term queries. A user study was performed to compare two variants of RecoOn with AK-TiveRank, in which users were shown two random lists to a query, for which they had to choose which one they prefer.

NCBO 2.0 [141] provides a ranking of ontologies and ontology sets based on a free text corpus or a list of keywords. The ranking fundamentally relies on an annotator that parses the input text and assigns matching terms from the ontology corpus to it. The annotated input is then used for scoring, which is based on four evaluation criteria: coverage (relevance), acceptance (importance), detail (relevance) and specialization (relevance). Each criterion is composed of several subscores, for which a fixed aggregation is proposed. However, the weights to combine the four evaluation criteria to determine the overall ranking score can be changed by the user. Apart from its detailed approach to scoring ontologies, a novelty is the possibility to retrieve a ranking of ontology sets. I.e., if no single ontology is able to cover the input, it makes sense to suggest the optimal combination of ontologies to increase the coverage. This is a challenging problem due to the high number of potential combinations (e.g., considering all 2- and 3-combinations for an ontology repository with size $|R|$ corresponds to $\binom{|R|}{2}+\binom{|R|}{3}$)). NCBO 2.0 reduces the complexity by applying heuristics, e.g., by removing ontology combinations that cover the exact same terms of the input. The ranking model is implemented as part of the BioPortal platform and provides a detailed scoring output to users. However, to the best of our knowledge, no user-based evaluation of the ranking model is provided. Moreover, choosing the right weights manually for the best combination of ranking features may be a difficult task for inexperienced users.

3.4 Ontology Ranking Evaluation

In the previously introduced ORTs and ontology ranking models, various evaluation approaches have been followed. Especially in the case of ORTs, however, the quality of the ranking model was often not evaluated. In this section, we present a brief overview of the historical developments of ontology ranking evaluation, the state-of-the-art, and its limitations.

Early works on ranking in the Semantic Web, such as **Swoogle** [64], lacked any evaluation of the ranking quality, which was soon identified as a general problem [189]. In the following years, authors presented diverse evaluations along with proposed ranking models, either based on small user studies or qualitative discussions on ranking outputs: **Ding et al.** [65] made a qualitative comparison between two ranking models and highlighted the need for relevance judgments in order to perform a formal evaluation; similarly, **Lei et al.** agreed that their evaluation is biased based on their qualitative assessment [130]. A first evaluation based on how well the ranking matches user judgments was presented in **AKTiveRank** [6] using the Pearson Correlation Coefficient. The evaluation includes a comparison with Swoogle's ranking, which resulted in a negative correlation with the ground truth (inverse ranking) and proved the lack of evaluations to be problematic. A similar evaluation approach has been followed later in **CARRank** [239]. Other evaluation approaches that were followed did not directly assess the ranking performance, but rather the tool's overall user experience based on a questionnaire, such as by **Falcons** [181], which requires extensive efforts and does not allow for immediate comparisons with newly proposed rankings. **Noy et al.** were the first to use more cost-efficient user feedback in the form of clicks to evaluate ontology rankings [156]. The evaluation is based on the Click Through Rate (CTR) on the first ranking position for several modifications of the ranking model over similar time intervals. However, the authors do not mine any relevance labels for the query-ontology pairs from the search logs. More recently published ontology ranking models still consider the evaluation as a challenging task. The complex **NCBO 2.0** recommender [141] is evaluated with a qualitative discussion based on absolute feature scores in comparison with a previous version of the NCBO recommender. The authors state that a user-based evaluation would help to understand the ranking's real-world effectiveness, but refrain from it since it would be a laborious task.

Despite the successful emergence of many ORTs over time in various domains, no attempts to unify ontology ranking evaluation known to us have been proposed until **CBRBench** [38] was released. To the best of our knowledge, CBRBench is the first published dataset with relevance labels for query-term pairs obtained from experts. It has further been used as input for LTR, such as in **DWRank** [37]. We consider CBRBench as the state-of-the-art in ontology evaluation, however, identify the following limitations.

- First, it only contains a small number of queries and judgments (10 and 819, respectively), meaning that its size limits its application in LTR settings.
- Second, it only compares a few ranking models individually (eight in total), and not a (potentially learned) combination of these or in configurations as they have been proposed in the literature. Moreover, it mostly compares ontology instead of term ranking features that assign the same score for all classes of the same ontology [38].
- Third, CBRBench is restricted to class ranking, meaning that it does not provide any judgments about properties, which limits its potential application to newly proposed ranking models.
- Fourth, the dataset might become outdated as soon as new ontologies arise and

relevance of terms changes, and because of its laborious way of obtaining the relevance labels, it cannot be easily updated.

Apart from CBRBench, other datasets have been proposed for LTR in the context of ontology ranking. In **TermPicker** [197], a dataset is automatically derived from the LOD cloud; however, relevance judgments only indicate whether a certain triple pattern appears in the LOD cloud or not (binary relevance) and it does not contain any judgments for keyword-based queries.

3.5 Summary

In this chapter, we present related work from the literature in the scope of ontology recommendation with regard to proposed tools, ranking models, and evaluation. In the next chapter, we develop a comprehensive taxonomy of respective approaches, present an evaluation, identify their limitations as well as open research challenges in more detail, and place these tools in the context of IoT ecosystems.

Part II

Formal Approach to Ontology Recommendation

Chapter 4

A Taxonomy and Evaluation of Ontology Recommendation Tools

This chapter aims to identify the limitations of existing Ontology Recommendation Tools (ORTs) (RQ 1). Moreover, we discuss the role and applicability of existing tools for the IoT, including respectively dedicated ontology collections (RQ 2). In order to identify gaps, we first develop an in-depth taxonomy of existing ORTs, which is used to classify and discuss previously presented related work. Second, we discuss the integration of ontology recommendation in the IoT ecosystem context and highlight shortcomings in the state-of-the-art. Based on these insights, we derive directions for future research that motivate the remaining chapters of this thesis. This chapter is based on the work that has been presented in the following paper:

- Niklas Kolbe, Sylvain Kubler, Jérémy Robert, Yves Le Traon, and Arkady Zaslavsky. "Linked Vocabulary Recommendation Tools for Internet of Things: A Survey". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 127. DOI: `10.1145/3284316`.

Contents

4.1 Introduction

Ontology recommendation is a composition of several processes, which themselves inherit various challenges. According to [204], the reuse of ontologies is divided into three aspects; discovery, selection, and integration (cf. Section 2.1.3). Furthermore, ontology recommendation is performed for a specific purpose or scenario [193], i.e., a recommendation could differ based on the user's intent of usage, which motivates the investigation of existing tools' applicability for the IoT. In general, such Semantic Web scenarios and tools include ontology-based query answering and semantic browsing [193], mapping and publishing data to the LOD cloud [198], ontology and knowledge engineering [204], as well as semantically annotating IoT data streams [84] as addressed in this dissertation.

Within this context, this survey aims at reviewing and assessing relevant tools (cf. Chapter 3) with regard to existing state-of-the-art theories, techniques and approaches. This evaluation is subsequently used as the basis for identifying and discussing challenges of the integration of ontology recommendation in IoT ecosystem tools. As of the time of writing, one may find related surveys of ontology recommendation and related work on architectural design considerations with respect to ontology discovery and/or selection [60, 193, 95, 235], as well as the integration of semantics in the IoT [203, 219]. However, to the best of our knowledge, no previous work has proposed a joint conceptualization nor an extensive framework to compare existing recommendation tools of various types with a similar purpose, nor reviewed the feasibility of such tools for IoT ecosystems. The term Ontology Recommendation Tool (ORT) is used as an umbrella term for tools that provide means for discovery and selection of ontologies.

The remainder of this chapter is structured as follows: The ontology recommendation taxonomy is developed in Section 4.2. The taxonomy serves as an evaluation framework for comparing existing ORTs in Section 4.3, which further presents the findings. Section 4.4 discusses the integration challenges of ontology recommendation in today's IoT projects and summarizes the identified research challenges and directions; the conclusion follows.

4.2 ORT Taxonomy Specification

A first overview of ORTs is presented in Section 2.1.3 and illustrated in Figure 2.2. In this section, we provide a complementing in-depth discussion of ontology recommendation by identifying key features for several dimensions. The taxonomy is structured into general features (Section 4.2.1), discovery features (Section 4.2.2), and selection features (Section 4.2.3). The enumeration of dimensions/features, as presented in the following, is kept throughout this chapter. Section 4.2.4 summarizes the final taxonomy.

4.2.1 General Features

Before going into detail for ontology discovery and selection, general dimensions and features are defined to characterize ORTs. Features are associated with two general dimensions, namely *approach* and *tool characteristics*, as summarized in Figure 4.1.

4.2.1.1 Approach

This first dimension is introduced to present the approaches, considering the following features:

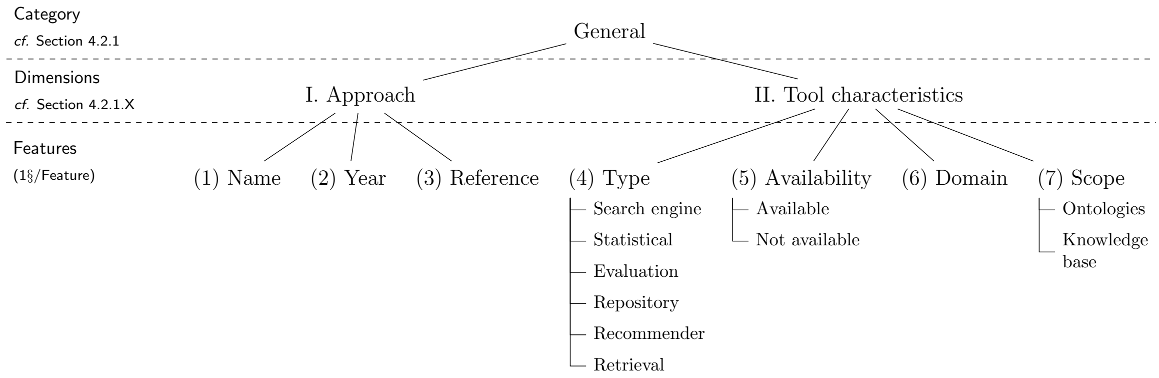


Figure 4.1: General dimensions and features to characterize ORTs.

Name: The name of the tool as used in its publication, to uniquely identify the approach.

Year: Year of the first associated publication that serves for identification of trends in proposed tools over time.

Reference: Scientific reference of the study used as basis for the evaluation.

4.2.1.2 Tool Characteristics

Secondly, more detailed characteristics of the tools are considered. These general features include:

Type: Different types of ontology libraries have been identified in [60]. However, as a broader scope of recommendation tools is considered, an own classification scheme was used that is based on the dimension(s) a tool puts particular emphasis on. Six different types of ORTs were identified, namely:

- *Search engine:* Focus on ontology collection, e.g., discovery and indexing of semantic documents through Web crawling.
- *Statistical tool:* Focus on ontology collection and evaluation, e.g., analyzing the usage of ontologies, often extracted from LOD sources, to guide end-users in choosing an appropriate ontology/term.
- *Evaluation tool:* Focus on ontology evaluation, e.g., assessing the quality of the discovered ontologies to give recommendations (considering a given set of metrics).
- *Repository:* Focus on ontology curation, e.g., the provision of a platform for a community to manually collect and review ontologies based on pre-defined requirements.
- *Recommender:* Focus on ontology ranking, e.g., by applying information filtering techniques or learning over LOD datasets to recommend the most suitable ontologies/terms.
- *Retrieval tool:* Focus on ontology interaction and matching, e.g., by proposing advanced means for querying, exploring, and matching candidates upon a query from an existing set of ontologies.

Availability: Whether the tool is *available* or *not available*[1], which also indicates

[1] Availability as of 09/2018

whether it could be evaluated experimentally. It is determined by checking whether an active website or download of the tool could be found by following URLs in the publication(s) and via a Web search with the tool's name.

Domain: Covered domains of the ontology collection (if not independent). It is concluded from the ontologies that are maintained in the tool's repository.

Scope: Indicates whether the approach focuses on *ontologies* or further supports *knowledge bases* (since schemas and data in the Semantic Web are based on the same formalism). The scope is inferred by checking whether the tool's repository exclusively contains ontologies.

4.2.2 Discovery Features

The discovery of ontologies is a fundamental process of ORTs, as only discovered ontologies can be in the set of potential candidates to be recommended upon a query. Three dimensions with regard to discovery are discussed, namely *collection*, *evaluation*, and *curation*, as summarized in Figure 4.2.

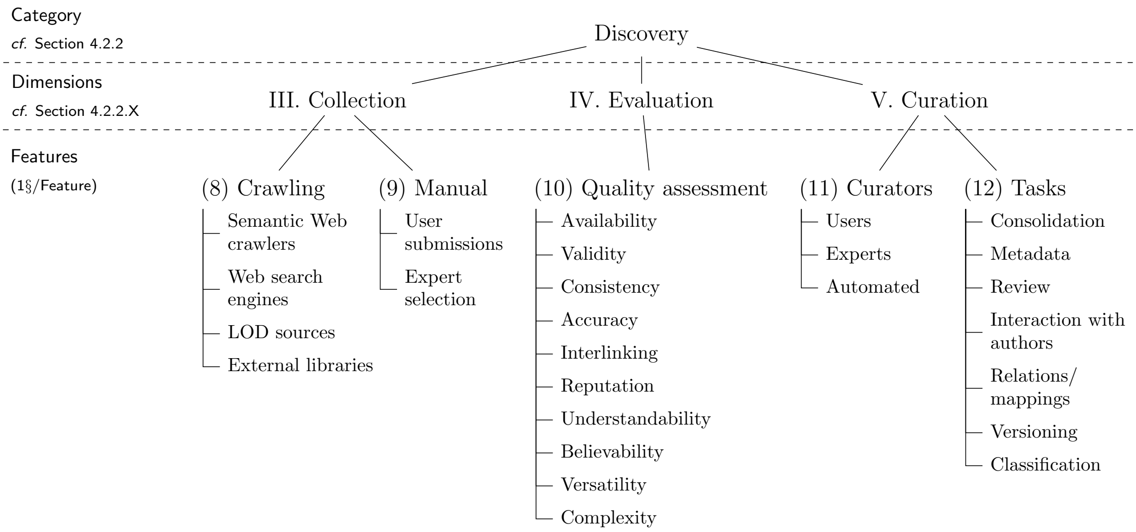


Figure 4.2: Dimensions and features related to ontology discovery.

4.2.2.1 Collection

The first step of the recommendation process is concerned with the collection process of available ontologies. Two distinct features and associated approaches were identified through the tools' evaluation and based on the discussion on ontology collections in [60], namely:

Crawling: In this process, the Web is browsed systematically by a software system, either realized through Semantic Web crawling, reusing common Web search engines, processing LOD sources and endpoints, or accessing APIs of existing ontology collections and ORTs. These approaches rely on a fundamental best practice that states that ontologies should be hosted and made publicly accessible at the URI of the ontology.

Semantic Web crawlers, also referred to as RDF crawlers, harvest data from Semantic Web Document (SWD) to discover ontologies or data. These crawlers focus on extracting RDF-based data that can be found in various formats (e.g., XML Syntax for RDF (RDF/XML), Turtle, JSON-LD), or embedded in other documents (e.g., RDF in Attributes (RDFa) in Hypertext Markup Language (HTML)). Existing ORTs also exploit conventional *Web search engines* and associated crawlers (Google, etc.) in order to discover SWDs on the Web (e.g., by filtering specific document types such as *filetype:rdf* and *filetype:owl*). Deploying an RDF crawler faces various design and implementation challenges, e.g., as discussed in [94]. As an alternative way to browsing the whole Web, accessing *LOD endpoints* or data dumps to extract used ontology terms is another way employed for building an ontology collection. However, this approach is only capable of discovering ontologies that have been used to model data in the analyzed datasets. Lastly, some approaches crawl ontologies from APIs of existing *external libraries*.

Manual: In contrast to automatic approaches, those who rely on a manual collection process do not aim to discover all available ontologies on the Web, but rather to fulfill one of the following goals: (i) present a proposed ontology to the community, (ii) keep supervised control over the maintained candidate set, or (iii) provide a platform for community consensus. Manual collection can be achieved either through *user submissions* or *expert selection*. Submission-based approaches are more flexible and facilitate the evolution of the ontology collection. Expert collections are often maintained by an official body.

4.2.2.2 Evaluation

The second dimension of ontology discovery is concerned with the assessment of the quality and correctness of ontologies [99]. In the ontology recommendation process, the purpose of evaluating ontologies are twofold: (i) assuring a certain quality for the selected ontology candidates; and (ii) giving the best recommendation for selection [193, 224]. Thus, as illustrated in Figure 2.2, evaluation serves as an input for *curation* as well as *ranking*, which are respectively discussed in Sections 4.2.2.3 and 4.2.3.3. In this review, the focus is on evaluation aspects relevant for ontology recommendation, rather than on ontology evaluation for the ontology engineering process, which has already been the subject of studies in the literature [99, 212, 27, 76].

In this respect, the Semantic Web community has proposed various best practices and guidelines for ontology design, development, publication and reuse. These documents often cover both schema- and data-related aspects, while providing a source for identifying quality criteria for ontologies. Well-known examples include the 5 star Linked Data model [22], consumer/publisher recommendations [103], Linked Data design considerations [96], five star rating for ontology use [110], OntoClean methodology [80], guidelines for Linked Data generation and publication [182], ontology pitfalls [176], W3C best practices recipes [24], or still the best practices applied to IoT [87]. Furthermore, quality assessment of ontologies has been extensively studied in the literature [67].

Quality assessment: The quality attributes considered in the taxonomy are listed in Table 4.1. The selection of criteria is mainly based on the comprehensive review presented in [247] and was complemented with those stemming from best practices and those considered by the ORTs that are subject of the evaluation. From the aforementioned sources, only quality attributes that are concerned with the schema (TBox) and deemed relevant for ontology recommendation were selected. In addition, it is shown whether the quality criteria are typically considered for curation and ranking.

4.2.2.3 Curation

The last identified dimension of the discovery process concerns ontology curation, which refers to the management and maintenance of the ontology candidates from

Table 4.1: Ontology Evaluation Criteria and Their Consideration in ORTs

Criteria	**Synonyms**	**Description**	**Implementation**	**Used for: Cur.**	**Rank.**
Availability [247, 22, 217]	Dereferencability [103, 96, 110] Accessibility [103]	Whether the vocabulary is accessible and dereferencable through its URI.	Hypertext Transfer Protocol (HTTP) requests, openness of vocabulary license.	✓	✓
Validity [178]	Syntactic accuracy/correctness [247, 22, 217] Syntax evaluation [212, 103]	Whether the vocabulary is syntactically correct.	Parsing the vocabulary.	✓	✓
Consistency [247, 99, 76, 103, 67]	Machine-readable [110]	Whether the vocabulary is free of logical contradictions with regard to its underlying representation (RDFS, OWL-variant, etc.).	Applying reasoners.	✓	✓
Accuracy [247, 99]	Domain cohesion [99, 212] Veracity [217] (Re-)usability [67]	Whether the schema correctly represents a real-world domain.	Human judgment.	✓	✓
Interlinking [247, 22, 96, 110]	Connectedness [31] Coupling [99] Structural evaluation [212]	The extent to which the vocabulary includes sufficient semantic relations to external vocabularies.	Counting in- and out-links at the schema level.	✓	✓
Popularity [193]	Usage statistics [212]	The extent to which the vocabulary/term is often used to model data of the domain it describes.	Analyzing LOD datasets for instantiations of the vocabulary, counting its presence in ontology repositories, or taking into account the number of local requests.	✗	✓
Reputation [247]	-	Whether users judge the vocabulary to be of integrity.	User reviews and ratings.	✓	✓
Understandability [247, 96]	Practical quality [217] Interpretability [247] Clarity [99] Metadata [212, 110]	Whether the vocabulary can be understood without ambiguity; e.g., through annotation properties like `rdfs:label` and `rdfs:comment`.	Counting annotation properties.	✓	✓
Believability [247]	Provenance metadata [96]	Whether the provenance / metadata about the vocabulary indicates that it comes from credible source.	Checking author information and history.	✓	✓
Versatility [247]	-	Whether the vocabulary is available in different languages and serialization formats.	Checking labels with language property (`@en` etc.).	✓	✗

Table 4.1: Ontology Evaluation Criteria and Their Consideration in ORTs (Cont.)

Criteria	Synonyms	Description	Implementation	Used for: Cur.	Rank.
Richness [67, 193]	Complexity [217] Density [6] Informativeness [10]	The extent to which concepts in the vocabulary are described and specified.	Measure based on number of properties, siblings, subclasses, and superclasses per concept.	✗	✓
Centrality [40]	Betweenness [6]	The extent to which a concept is central in the vocabulary graph.	Measure based on amount of relations of a concept and/or the count of shortest paths within the vocabulary that go through it.	✗	✓
Importance [64]	-	A combination of popularity and interlinking, meaning that the importance depends on the quality of the link.	Measures of interlinking while taking into account the popularity of the source for in-links, e.g., PageRank algorithm.	✗	✓

the internal repository. Indeed, curation is often a collaborative effort to ensure and improve the quality of formalized knowledge [78]. Two features were identified based on the survey in [78] and the reviewed tools in this respect, namely:

Curators: This feature indicates who oversees the curation and maintenance of the ontology candidate collection, which could be fulfilled by *users*, *experts* or through *automated processes*. Whereas human-based curation offers means to improve ontologies based on reviews and discussions, automated curation is able to handle large sets of discovered ontologies more efficiently.

Tasks: The curation process can cover different aspects, including *metadata* completion and maintenance, *review* of newly discovered ontologies, add *semantic relations* to other ontologies of the corpus, *consolidation* of discovered ontologies, support of *versioning*, as well as *classification*. It should be noted that the curation process strongly influences the quality and extent of the ontology candidate repository, which could potentially enhance ontology selection (e.g., the classification of ontologies can be exploited to filter domains).

4.2.3 Selection Features

The second fundamental task of ontology recommendation is to select the most appropriate candidate from the repository. The dimensions with regard to selection are *interaction*, *matching*, and *ranking*, as summarized in Figure 4.3 and discussed in the following.

4.2.3.1 Interaction

Recommendation approaches need to provide means to interact with users and agents to query the recommendation service. The dimension is broken down to the following features:

Query format: An ORT interface could offer the following means/formats for querying, as identified from evaluated tools (in particular from [91]) and the discussions in

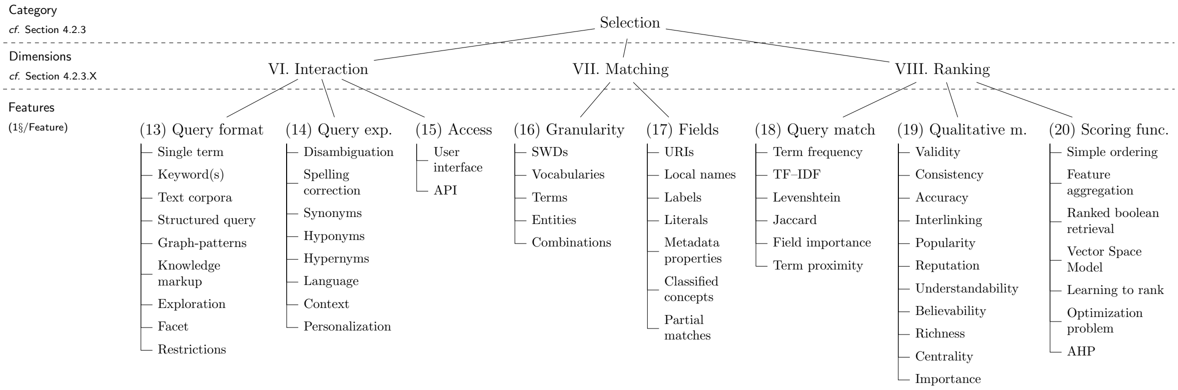


Figure 4.3: Dimensions and features related to ontology selection.

[97]: *single terms*, *keyword*-based search, *text corpora* (so-called free text retrieval), *logical/structured query* (e.g., SPARQL), *graph-patterns*, and tasks expressed with *knowledge markup*. Whereas keyword-based search is the most popular and easy to use interface, it is often argued that it does not allow for a precise specification of the information need [97]. Search- and recommendation-based approaches often require multiple interactions for users to reach their goal, thus, further employed interaction models include *exploration* of the ontology collection (e.g., object focus, path traversal [91]), specification of *restrictions*, and offering *facets*.

Query expansion: The idea of query expansion is to express the user's information need more precisely for enhanced retrieval results. Available approaches, based on the comprehensive survey in [42], include *disambiguation* options (e.g., defining the sense of a keyword in case it has multiple meanings [242, 83]), perform *spelling checks* on the user's input, expend keywords with *synonyms*, *hyponyms* or *hypernyms* (e.g., through WordNet [148]). Some approaches aim to *personalize* the query and retrieve more suitable results for a specific user. In semantic search, such user preferences could represent the interest of the user in a certain concept. Instead of explicitly defining preferences, another approach consists of taking into account *contextual parameters* of a request to improve the recommendation.

Access: ORTs are often designed to provide search that is oriented towards humans, and thus often offer a *User Interface (UI)* to access the information. In more detail, yet out of the scope of this survey, UIs are concerned with result presentation and visualization (list, graphs, trees, etc.). However, the need for Semantic Web applications to discover ontologies lead to the need for specifying *APIs* (typically following Representational State Transfer (REST) principles) to access the service, which further allows to aggregate several ORTs.

4.2.3.2 Retrieval

The second dimension of ontology selection, retrieval, is concerned with:

Granularity: The importance of granularity for ontology recommendation is discussed in [193, 186] and served as motivation to collect granularity levels that were considered by the evaluated tools. Retrieval in ORTs can be done on the level of matching *SWDs*, *ontologies*, *terms*, *entities*, or *combinations* of different ontologies. The granularity strongly impacts the nature of the recommendation process: Whereas some approaches aim at recommending complete ontologies that have the best coverage of the queried domain/concepts (also through recommending combinations of ontologies), other approaches seek to find a single best term or entity. Indeed, it is not trivial whether it is best to reuse as little as possible number of ontologies for a user's intended task or rather combine "better" terms from various ontologies [196].

Matching fields: The matching process is concerned with retrieving candidates from the repository that match the query. This feature shows the detail to which tools match a query against the information contained in an ontology. Due to the inherited structure of RDF ontologies, matches can be performed on different fields and properties, including *URIs*, *local names*, *labels*, *literals*, and *metadata properties* such as basic properties like `rdfs:comment` or properties from ontologies such as the Dublin Core[2] and SKOS[3]. Moreover, in case ontologies are classified during the curation process, ontologies of a *matching domain/concept* can be retrieved. In the case of logical queries, the match is returned from processing the query based on its underlying language. Furthermore, *partial matches* could also be taken into account, even though one may dispute its real impact on the search quality [113]. The matching fields considered by an approach are relevant for ontology selection, since there is no one way or guarantee that ontologies are annotated in the same way. Valuable information could reside in different fields/properties that could help to identify a matching candidate.

4.2.3.3 Ranking

Algorithms to rank matched candidates form a key component of ORTs. Ranking aims at determining best candidates of the ontologies that matched the query by taking into account various measures, including:

Query match measures: These enable the ranking of the candidate set determined by the query match through content- and graph-based similarity measures. Due to the huge amount of similarity measures in the literature, the taxonomy is limited to those found in the evaluated tools. However, these were aligned to the literature, such as [139, 38]. A common approach is to compute the *Term Frequency (TF)* in an ontology for all words in a query, which is often combined with the Inverse Document Frequency (IDF) of the terms (otherwise, rare terms would have no power to influence the query relevancy), also known as *TF–IDF* [194]. Another way to calculate the similarity between query and candidate, at the string level, is to compute the edit distance (*Levenshtein*). The *Jaccard coefficient* allows to measure the overlap between sets, and thus can be applied to the set of words from the query and those from the ontology. Further, finding a match in some field types of an ontology might be of higher importance than others (e.g., a match in the name is more valuable than in the metadata), which can be represented through assigning different weights to field types (i.e., *field importance* or weighted zone ranking [139]). The last query match measure found is concerned with the *proximity* of multiple query term matches within the ontology graph. This measure is calculated by identifying the shortest path between matched fields [139].

Qualitative measures: These aim to compute a score for an ontology or term independent from the query. The approach to the collection of qualitative features is discussed in Section 4.2.2.2 and those quality criteria relevant for ranking are indicated in Table 4.1. One scoring algorithm standing out is PageRank [164] to measure the importance of a document, a popular one for ranking Web pages that can be adopted

[2] Dublin Core ontology: http://purl.org/dc/terms/ – accessed 09/2018

[3] SKOS ontology: https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html – accessed 09/2018

to the needs for SWDs. PageRank falls into link analysis, being based on a random surfer who, starting from one page/ontology, randomly follows a link. The more often a node is visited by the random surfer during his walk, the more important it is. For SWDs, the random surfer needs to consider the semantics of the followed links.

Scoring functions: Lastly, previously presented measures are used as inputs of the scoring function to compute an overall ranking of matching candidates for a query. Due to the large number of approaches to achieve a ranking, only those found in the evaluated tools are listed in the taxonomy. Further reading includes general scoring functions [139] and ranking of ontologies [215, 45].

Ranking with only one feature requires *simple ordering*. In a straightforward manner, multiple features can be *aggregated*, e.g., through weighted or unweighted sum or factorization. In the *Vector Space Model (VSM)* [195], documents are represented as vectors with each component representing a document term, which could be computed based on the TF–IDF-like measures. Vector space scoring calculates the similarity between two documents (e.g., between a query vector and an ontology vector) by calculating the cosine similarity, which, however, is expensive to compute [139]. Weights of ranking features can be defined through experts or learned from a training data set (*learning to rank* [132]), with algorithms such as LambdaMART [240]. Calculating a score for candidates could also be seen as an *optimization problem* by formulating features as cost functions. Lastly, features could also be aggregated through the *AHP* [192].

4.2.4 Taxonomy Summary

The overall taxonomy for ORTs consists of the different sets of features with regard to the different ontology recommendation dimensions, which have all been summarized in Table 4.2. This framework is then used in the following as a basis for evaluating and comparing various recommendation tools.

4.3 Evaluation of ORTs

This section presents the evaluation of existing ORTs based on the proposed taxonomy. Table 4.3 and 4.4 respectively present the results with regard to the general/discovery and selection dimensions. In the following, the findings of the evaluation are discussed.

To help readers extract quick and meaningful information, some results of the evaluation (i.e., from Table 4.3 and 4.4) have been displayed in the form of charts in Figure 4.4, including:

- Figure 4.4a: number of tools of each type that have been introduced and are still available;
- Figure 4.4b: how often evaluation criteria were used, and whether it is for quality assessment or ranking;
- Figure 4.4c: whether tools focus on quality assessment, ranking, or both;
- Figure 4.4d: to show which ranking features were used for each matching granularity;
- Figure 4.4e: trends of how often interaction features are used.

Shift from semantic search engines and evaluation to repositories, recommenders and retrieval: One trend that can be observed is the shift from the

Table 4.2: ORT Taxonomy

Cat.	Dimension	Feature	Description
General	I. Approach	(1) Name	Tool's name.
		(2) Year	Year of first relevant publication.
		(3) Reference	Reference to the tool's academic publication.
	II. Tool characteristics	(4) Type	The category the tool falls into: Search Engine (■), Statistical (■), Evaluation (■), Repository (■), Recommender (■), Retrieval (■).
		(5) Availability	Whether the tool is available (✓), not available (✗).
		(6) Domain	Domains covered, if not domain-independent.
		(7) Scope	Whether tool focuses on Ontologies (Ont), or on Knowledge Bases (KB).
Discovery	III. Collection	(8) Crawling	Automatic collection of ontologies through: Semantic Web crawler (Cr), Web search engines (SE), LOD Endpoints (LOD), and/or External Libraries (Lib).
		(9) Manual	Manual collection through: User Submission (U), and/or Expert Collection (E).
	IV. Evaluation	(10) Quality assessment	Assessment of discovered ontologies for curation purposes, based on: Availability (Ava), Validity (Val), Consistency (Con), Accuracy (Acc), Interlinking (Int), Reputation (Rep), Understandability (Und), Believability (Bel), Versatility (Ver), or Richness (Rich).
	V. Curation	(11) Curators	Curation handled Automatically (A), by Experts (E), or Peers/Users (U).
		(12) Tasks	Curation tasks covered: Consolidation (Con), Metadata (Met), Content Review (Rev), Interaction with authors (Int), defining Relations and Mappings (Rel), maintain Versions (Ver), add Classifications (Clas).
Selection	VI. Interaction	(13) Query format	Ways to query the recommendation service, including: Single Term (Term), Keywords (Key), Text Corpora (TC), Structured (QL), Graph-pattern (Gra), Knowledge Markup (KM), Exploration (Exp), Restrictions (Res), Facets (Fac).
		(14) Query expansion	Means to improve the query formulated by the user: Disambiguation (Dis), Spelling Correction (Spe), Synonyms (Syn), Hyponyms (Hypo), Hypernyms (Hyper), Language (Tra), Context (Con), or Personalization (Per).
		(15) Access	Whether information access is provided for Users (UI), and/or Agents (API).
	VII. Matching	(16) Granularity	To which granularity a query is matched and retrieved from the corpus: Ontology (Voc), Terms (Term), Entities (Ent), SWDs (SWD), or support of Combinations of these (Comb).
		(17) Fields	To which fields of an ontology a query is matched: URIs (URI), Local names (Nam), Literals (Lit), Labels (Lab), Metadata Properties (Met), Classified Concepts (Con), Partial Matches (Par).

Table 4.2: ORT Taxonomy (Cont.)

Cat.	Dimension	Feature	Description
	VIII. Ranking	(18) Query match measure	Assessment of the similarity between query and ontologies in the corpus, based on: Term Frequency (TF), TF-IDF (TF-IDF), Levensthein (Lev), Jaccard (Jac), taking into account the Field Importance (FI), and/or the Term Proximity (TP) of matches in an ontology.
		(19) Qualitative measures	Assessment of ontologies in the corpus to calculate a quality score based on: Validity (Val), Consistency (Con), Accuracy (Acc), Interlinking (Int), Popularity (Pop), Reputation (Rep), Understandability (Und), Believability (Bel), Richness (Rich), Centrality (Cen), or Importance (Imp).
		(20) Scoring function	Approach to calculate a final rank based on the used measures: Simple Ordering (Ord), Feature Aggregation (Agg), Ranked Boolean Retrieval (RBR), Vector Space Model (VSM), Learning to Rank (LTR), Optimization Problem (Opt), Analytical Hierarchy Process (AHP).

Table 4.3: Evaluation of ORTs Regarding General and Discovery Features

I. Approach			**II. Tool characteristics**				**III. Collection**		**IV. Evaluation**	**V. Curation**	
(1) Name	(2) Year	(3) Reference	(4) Type	(5) Availability	(6) Domain	(7) Scope	(8) Crawled	(9) Manual	(10) Quality assessment	(11) Curators	(12) Tasks
Ontokhoj	2003	[167]	■	✗	-	Ont	Cr	-	-	A	Clas
OntoSelect	2004	[31]	■	✗	-	Ont	Cr	U	-	-	-
Swoogle	2004	[64]	■	✓	-	Ont, KB	Cr, SE	U	Val, Int	A	Met, Rel, Ver
Ontosearch	2005	[248]	■	✗	-	Ont, KB	SE	-	-	-	-
SWSE + ReConRank	2007	[93]	■	✗	-	Ont, KB	Cr	-	-	A	Con, Rel
Sindice	2007	[229]	■	✗	-	Ont, KB	Cr	U	Val, Con	-	-
Watson	2007	[57]	■	✓	-	Ont, KB	Cr, SE, Lib, *Swoogle*	-	Val, Con, Rep	A	Con, Rel, Met, Clas
Falcons Concept & Entity Search	2009	[46, 181]	■	✗	-	Ont, KB	Cr	-	Val	A	Clas
VisiNav	2009	[92]	■	✗	-	Ont, KB	Cr, LOD	-	-	-	-
WebOWL	2012	[19]	■	✗	-	Ont, KB	Cr	-	Val, Con	-	-
LODstats	2012	[12]	■	✓	-	Ont, KB	LOD	-	-	A	Met
vocab.cc	2013	[208]	■	✓	-	Ont	LOD	-	-	-	-
OUSAF	2015	[9]	■	✗	-	Ont	*Watson, Sindice*	-	Val, Con, Int, Und	-	-
Supekar et al.	2004	[217]	■	✗	-	Ont	*Onthokoj*	-	-	A	Clas
OntoMetric	2004	[133]	■	✗	-	Ont	U	-	-	-	-
Ontology Auditor	2005	[34]	■	✗	-	Ont	Lib	-	Val, Con, Acc	-	-
OntoQA	2005	[223]	■	✗	-	Ont, KB	*Swoogle*	-	Rich, Int, Und	-	-
Knowledge Zone + TS-ORS	2006	[131, 218]	■	✗	Biomed.	Ont	-	U	Acc, Und, Rep, Bel	U	Clas, Ver, Rev, Met

Table 4.3: Evaluation of ORTs Regarding General and Discovery Features (Cont.)

I. Approach			II. Tool characteristics				III. Collection	IV. Evaluation		V. Curation	
(1) Name	(2) Year	(3) Reference	(4) Type	(5) Availability	(6) Domain	(7) Scope	(8) Crawled	(9) Manual	(10) Quality assessment	(11) Curators	(12) Tasks
Open Metadata Registry	2006	[98]	■	✓	-	Ont	-	U	-	U	Ver, Met
Ontosearch2	2006	[166]	■	✗	-	Ont	-	U	Val, Con	-	-
Oyster	2006	[165]	■	✓	-	Ont	-	U	-	U	Met
OBO Foundry	2007	[206]	■	✓	Biomed.	Ont	-	U	Ava, Int, Und	U	Met, Ver, Clas, Rev
BioPortal	2009	[157]	■	✓	Biomed.	Ont	-	U	Int, Acc, Und	U	Rev, Met, Ver, Rel
Cupboard	2009	[58]	■	✗	-	Ont	-	U	*TS-ORS*	U	*TS-ORS*, *Oyster*, Rel
MMI	2009	[191]	■	✓	Marine	Ont	-	U	Val, Con	U	Met, Rel, Ver
Ontobee	2011	[241]	■	✓	Biomed.	Ont	*OBO Foundry*	E	-	-	-
BiOSS	2010	[142]	■	✗	Biomed.	Ont	-	E	-	-	-
Manchester OWL Repository	2014	[143]	■	✓	-	Ont	Cr, SE, Lib	U	Ava, Val, Con	A	Con
smartcity .linkeddata.es	2014	[175]	■	✓	IoT	Ont	-	E, U	Ava	E	Int, Met, Rev
LOV	2014	[232]	■	✓	-	Ont	-	U	Ava, Val, Und, Int, Bel	E, A	Met, Rev, Ver, Int
Ontology Lookup Service	2015	[114]	■	✓	Biomed.	Ont	-	U	Val, Con	E, A	Ver
Ontohub	2017	[50]	■	✓	-	Ont	-	U	Val, Con	U	Rev, Ver, Met, Rel
(Web)CORE	2006	[72, 41]	■	✗	-	Ont	Lib	U	Acc, Und, Rep	U	Rev, Clas
DWRank	2014	[40, 37]	■	✗	-	Ont	Lib	-	-	-	-
TermPicker	2016	[197]	■	✗	-	Ont	LOD	-	-	-	-

Table 4.3: Evaluation of ORTs Regarding General and Discovery Features (Cont.)

I. Approach			II. Tool characteristics				III. Collection	IV. Evaluation		V. Curation	
(1) Name	(2) Year	(3) Reference	(4) Type	(5) Availability	(6) Domain	(7) Scope	(8) Crawled	(9) Manual	(10) Quality assessment	(11) Curators	(12) Tasks
NCBO 2.0	2017	[141]	■	✓	Biomed.	Ont	*BioPortal*	-	-	-	-
AKTiveRank	2006	[6]	■	✗	-	Ont	*Swoogle*	-	-	-	-
(combi)SQORE	2007	[230, 231]	■	✗	-	Ont	*Watson*	-	-	-	-
LOVR	2015	[211]	■	✓	-	Ont, KB	*vocab.cc, LOV*		-	-	-
RecoOn	2016	[39]	■	✓	-	Ont	Lib	-	-	-	-

Table 4.4: Evaluation of ORTs Regarding Selection Features

I. Approach	**VI. Interaction**			**VII. Matching**		**VIII. Ranking**		
(1) Name	(13) Query format	(14) Query expansion	(15) Access	(16) Granularity	(17) Fields	(18) Query match measure	(19) Qualitative measure	(20) Scoring function
Ontokhoj	Term, QL	Con, Dis, Syn, Hyper	UI, API	Voc	Nam	-	Imp	Ord
OntoSelect	KM, Exp	-	UI	Voc	Lab, Par	TF	Int, Rich	Agg
Swoogle	Key, QL, Fac	-	UI, API	SWD, Term	Lab	TF-IDF	Imp	Ord
Ontosearch	Key	Per	UI	SWD	Con	TF-IDF	-	VSM
SWSE + ReConRank	Key, Fac	-	UI, (API)	Ent	Nam, Lab, Met, Lit	TF-IDF	Imp	Ord
Sindice	Key	-	UI	Ent	URI, Nam, Lab	TF-IDF	Ava	Agg
Watson	Key, Exp, QL	-	UI, API	Voc, Ent, Comb	Nam, Lab, Met, Lit, Par, Con	-	Rich	-
Falcons Concept & Entity Search	Key	-	UI	SWD, Voc, Ent	Nam, Lab, Lit	TF-IDF	Pop	VSM
VisiNav	Key, Fac, Exp	-	UI	Ent	Nam, URI, Lab, Lit	-	Imp	Ord
WebOWL	QL	-	UI, API	Ent	-	-	Imp	
LODstats	Term, Exp, QL	-	UI	Voc, Term, Ent	Nam, URI,	-	Pop	-
vocab.cc	Key	-	UI, API	Term	Nam, URI, Lab	-	Pop	Ord
OUSAF	QL	-	UI	Term	-	-	Pop, Rich	Agg
Supekar et al.	-	-	*Ontokhoj*	Voc	Con	-	Ava, Val, Acc, Rich	Agg
OntoMetric	-	-	UI	Voc	-	-	Ava, Acc, Und, Rich	AHP
Ontology Auditor	Exp	Syn, Hyper, Hypo	UI, API	Voc	Con	-	Val, Und, Rich	Agg

Table 4.4: Evaluation of ORTs Regarding Selection Features (Cont.)

I. Approach	**VI. Interaction**			**VII. Matching**		**VIII. Ranking**		
(1) Name	(13) Query format	(14) Query expansion	(15) Access	(16) Granularity	(17) Fields	(18) Query match measure	(19) Qualitative measure	(20) Scoring function
OntoQA	Key	Syn	UI	Voc	Nam	-	Int, Pop, Rich, Cen	Agg
Knowledge Zone + TS-ORS	Key, Exp	Per, Hypo	UI	Voc	Nam, Met, Con	-	Acc, Rich, Rep, Bel	Agg
Open Metadata Registry	Key, Exp, QL	-	UI	Voc, Term	Lab, Con, Met	-	-	Ord
Ontosearch2	Key, QL, Res	Syn	UI	Term, Ent	URI, Nam, Lab	TF, FI	-	Agg
Oyster	Term	-	UI	Voc	Nam	-	-	-
OBO Foundry	Exp	-	UI	Voc	-	-	-	-
BioPortal	Key, Exp, Fac	Syn	UI, API	Voc, Term	Nam, Lit	-	-	-
Cupboard	*Watson*	-	UI	Voc	*Watson*	*TS-ORS*	-	*TS-ORS*
MMI	Key, Fac, Exp, QL	-	UI, API	Voc, Term	-	-	-	Ord
Ontobee	Key, Exp, QL	-	UI	Voc	Lab, Par	-	-	Ord
BiOSS	Key	Dis, Syn	UI, API	Voc, Comb	Nam, Con, Lab, Par	TF	Rich, Pop	Agg
Manchester OWL Repository	QL, Exp	-	UI, API	Voc	-	-	-	-
smartcity .linkeddata.es	Term, Exp	-	UI	Voc	Con	-	-	Ord
LOV	Key, QL	-	UI, API	Voc, Term	Nam, Lab, Met	TF-IDF, FI	Pop	Agg
Ontology Lookup Service	Key, Exp, QL	-	UI, API	Voc, Ent	URI, Nam, Lab, Par	-	-	-

Table 4.4: Evaluation of ORTs Regarding Selection Features (Cont.)

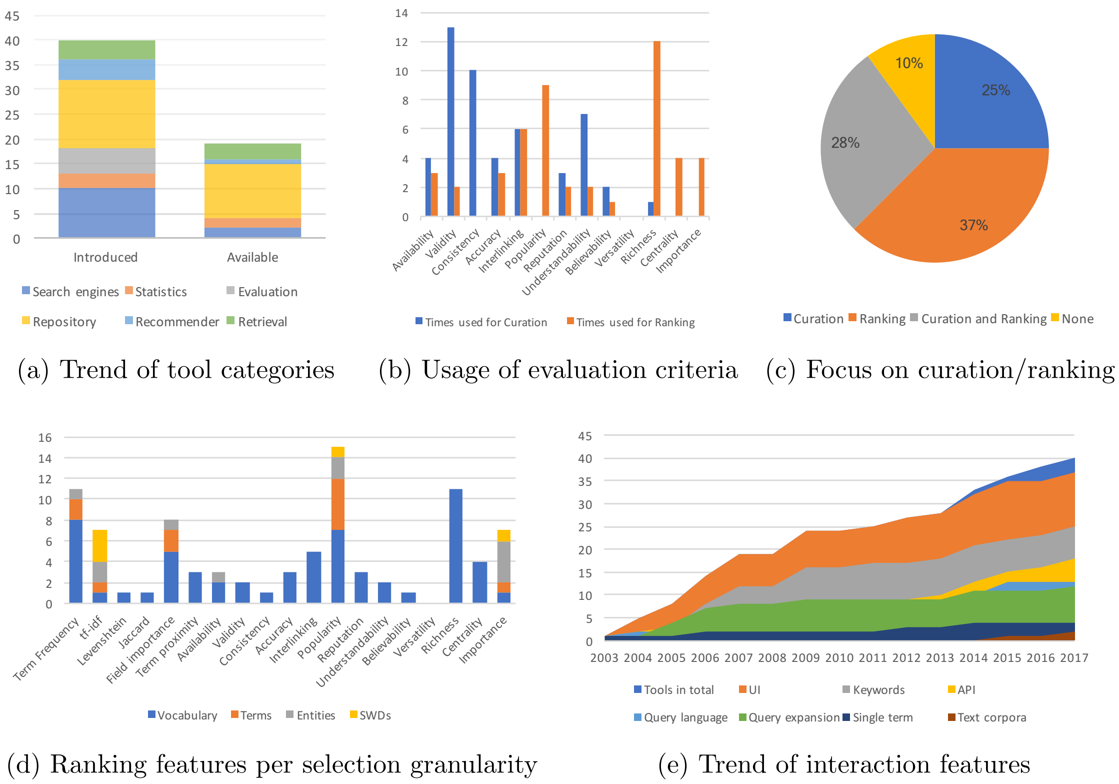
I. Approach	**VI. Interaction**			**VII. Matching**		**VIII. Ranking**		
(1) Name	(13) Query format	(14) Query expansion	(15) Access	(16) Granularity	(17) Fields	(18) Query match measure	(19) Qualitative measure	(20) Scoring function
Ontohub	Key, Exp	-	UI, API	Voc	URI, Nam, Met	-	-	-
(Web)CORE	Key	Per, Dis	UI	Voc	Nam, Par	TF, FI, Lev	Rep	VSM
DWRank	Key	Syn	-	Voc	Nam, Met, Par, Con	TF, FI	Int, Cen	LTR
TermPicker	Gra	-	UI	Term	Nam	-	Pop	LTR
NCBO 2.0	Key, TC	Syn	UI, API	Voc, Comb	Met	TF, FI	Pop, Rich, Cen	Agg
AKTiveRank	Key	-	UI	Voc	URI, Lab, Par	TF, FI, TP	Int, Cen	Agg
(combi)SQORE	Key, Res	Syn, Hyper, Hypo	UI, API	Voc, Comb	Nam	TF, TP	Rich	Agg
LOVR	TC	-	UI, API	Term	*LOV, vocab.cc*	*LOV, vocab.cc*	*LODStats*	Agg
RecoOn	Key	-	API	Voc	Nam, Par	Jac, TP	Int, Pop, Rich	Opt

development of search engines and evaluation-focused tools to recommender and retrieval systems. As depicted in Figure 4.4a, only a few search engines and evaluation tools included in the survey are still available (about 50% of the reviewed ORTs). Even though a similar observation can be made about tools of type recommenders, it should be noted that three out of four tools from this category have been introduced just in the recent years, indicating a growing interest. Semantic search engines are often not solely focused on ontologies but also on Linked Data, however, conventional Web search engines like Google increasingly incorporate capabilities of retrieving Semantic Web content, as claimed by [153] – a well-known example is the schema.org ontology embedded in websites which is supported by many conventional Web search engines. Evaluation tools are able to thoroughly assess ontologies; however, they are often inefficient in finding suitable candidates from an ontology reuse standpoint. Indeed, with the huge amount of ontologies published on the Semantic Web, the challenge lies not in discovering as many as possible, but rather in selecting efficiently as few as possible well-fitting and requirements-meeting ontologies/terms.

Curation vs. ranking: A fundamental aspect of recommendation is the assessment of ontologies' quality. Figure 4.4b shows how often the evaluation criteria are used for curation and ranking. Most curation approaches focus on ensuring validity (13 times), consistency (10) and understandability (7) of newly collected ontologies, whereas ranking models rather take into account the richness (12), popularity (9), and interlinking degree (6). Considering Figure 4.4c, it can be added that ORTs often focus on either the curation of the ontology collection (25%) or efficient ranking for queries (37%). However, the combination of both, which is implemented by 28% of the reviewed ORTs, would naturally increase the quality of the recommendation service. An example thoroughly considering both approaches is the LOV platform [232].

Limited support of combined recommendations: As previously mentioned, it is not trivial whether a recommendation should be made on an ontology or term/entity level. When publishing IoT data, rarely a single ontology would cover all required terms. The most common selection granularity of the surveyed tools is a complete ontology. Identifying the combination of most suitable and interlinked terms for an existing non-linked schema cannot be easily achieved with existing ORTs, as engineers are still required to pick and combine suitable terms. Combined recommendations are especially useful when the recommendation service can take into account all the terms (or other data structures) that the user/agent is looking for. In this evaluation, only two approaches offer recommendations based on text corpora, namely NCBO 2.0 and LOVR, whereas the latter takes HTML as input with the goal to semantically annotate websites.

Impact of qualitative measures on ranking quality remains unclear: Despite the various evaluations for recommendations presented in the selected studies, the general importance of qualitative measures to achieve a better quality of ranking remains unclear, as most approaches focus on a limited set of criteria, and different metrics are used for same criteria. The selection approach by Semantic Web users is often driven by the popularity of an ontology, as claimed in [196]. The evaluation reveals that popularity is also among the most used criteria for selection in the surveyed ORTs (cf. Figure 4.4d). However, in order to receive good results these query-independent features need to be combined at least with a reliable query match measure [38]. In general, establishing the correct weight between features to optimize the ranking model is a tedious task [132], and there is no common conclusion on the importance of each rank-



(a) Trend of tool categories (b) Usage of evaluation criteria (c) Focus on curation/ranking

(d) Ranking features per selection granularity (e) Trend of interaction features

Figure 4.4: Analysis of the ORTs evaluation.

ing feature for ranking models. Only two approaches of the survey use learning-based approaches to assign weights to features, namely TermPicker that focuses on different metrics related to popularity, and DWRank that focuses on learning the weights for features like centrality and importance. Furthermore, the aggregation of features is also dependent on the selection granularity. Figure 4.4d shows the features used for ranking per selection granularity. It can be observed that a large variety of features is only considered (and suitable) when ranking complete ontologies/SWDs, whereas only a limited number of features is used for ranking terms/entities.

Simplicity for interaction: The trend of some interaction features is displayed in Figure 4.4e. It can be observed that simple interfaces are more popular. Most approaches are keyword-based and increasingly offer APIs. On the other hand, the use of query expansion features to refine queries for users and alternative query formats (e.g., query languages or text corpora) are less often implemented by the surveyed ORTs.

4.4 Discussion

In this section, we propose the conceptual integration of ontology recommendation in IoT ecosystems in Section 4.4.1, present the consideration of ORTs in today's IoT platforms in Section 4.4.2, and conclude research challenges in Section 4.4.2.

4.4.1 Conceptual Integration of ORTs in IoT Ecosystems

As previously mentioned, the process of recommending ontologies can be structured in three steps, namely (1) discovery of available ontologies, (2) selection of the most suitable candidates for user queries, and (3) integration of the recommendation for the

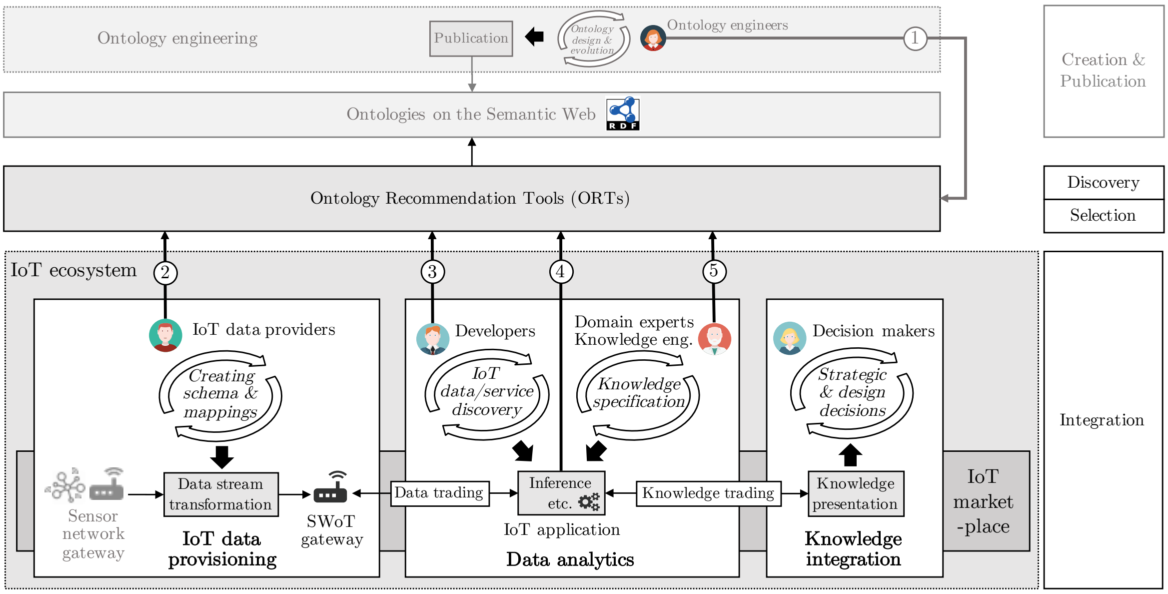


Figure 4.5: Integration of ontology recommendation in the IoT ecosystem model.

user's task at hand. Figure 4.5 illustrates the *integration* of ORTs in the scope of an IoT ecosystem model, which is dependent on the user's intent.

Five distinct use cases (denoted by ① – ⑤ in Figure 4.5) provide insight into different users/agents who query ontology recommendation for different purposes. Case ① shows the most fundamental use case from the Semantic Web, in which ontology engineers use recommendation tools during the ontology development process in order to link to, and potentially extend already existing definitions during the development process instead of redefining them. Cases ② – ⑤ show integration cases in the IoT ecosystem model. ② shows the case of smart object owners, who use the recommendation not only to define a semantic schema, but also to create mappings from local sensor data to the newly defined schema. With these mapping rules, sensor data streams can be transformed and published with semantic annotations, which is a core requirement to efficiently join IoT ecosystems (e.g., to be easily and efficiently discovered). Cases ③ and ⑤ represent queries from developers and domain experts, who respectively intend to discover available IoT data/services and specify knowledge for an intelligent IoT application. Some of these processes could also be automated through artificial agents requesting for ontology recommendation (④). Eventually, the ontology recommendation fosters interoperability and allows for more efficient and elaborate knowledge extraction.

4.4.2 Consideration of ORTs in IoT Platforms

As thoroughly reviewed in [149], existing open IoT platforms lack unified and interoperable data models. Ongoing analysis, discussions and IoT project efforts with regard to ontologies for the IoT domain indicate that this issue is thoroughly addressed by the community, as evidenced by [14, 73]. Despite the fact that more and more IoT platforms support Semantic Web technologies, and that ontologies to describe *Things* are becoming more mature, solely the use of ontologies is not enough to achieve global interoperability [18]. As a first step, this only makes those platforms interoperable that either use the same ontologies, or ontologies that have been successfully mapped/matched. Recent efforts in this regard include, e.g., the Fiesta IoT project that achieved semantic interoperability between the FIWARE and OneM2M platforms (both using different data models/ontologies) [128]. Before, the SPITFIRE project was concerned with aligning IoT ontologies [174].

ORTs form a key building block to support users of semantic-aware IoT platforms, for all the IoT ecosystem use cases that have been previously introduced in Section 4.4.1 (i.e., linked sensor data publication/transforming sensor data streams, discovering IoT data/services, defining domain knowledge). However, it can be observed that existing IoT platforms – despite the support of Semantic Web technologies – do not yet follow the IoT ecosystem model, and do not consider ontology recommendation in their scope of tools and platforms. One reason that could explain this is that ontologies for IoT-related domains (mobility, city, home, etc.) have not yet reached full maturity, many still being under specification and development (such as MobiVoc). However, the expectation is that developers can easily extend platforms to their needs, integrate data from and model data in a format that is understood by various platforms [149]. ORTs, in their essence, support this goal through collecting and offering means for selecting appropriate ontologies. The recent and promising Industrial Ontologies Foundry (IOF) initiative[4], to some extent, agrees with this vision and aims to adapt the success story of the OBO Foundry from the medical domain to the industrial domain (including IoT), in order to provide a collaborative tool suite that helps to build and collect jointly interoperable ontologies. Nonetheless, the idea of sharing and reusing data models defined by the community has already found its way to the IoT, e.g., the information model repository (based on a domain-specific language) of the Eclipse Vorto tool[5].

Despite the lack of consideration of ORTs in IoT platforms, they have been considered in other ontology-based tools. For example, [197] describes the integration of TermPicker in Karma [118], which is a linked data integration tool based on mapping rules. Still, such tools do not satisfy IoT specific requirements, e.g., applying the transformation on data streams, while considering specific characteristics of sensor data streams [17], and publishing the data in a SWoT gateway. On the other hand, IoT-specific tools often do not consider ontology recommendation. Instead, they are built upon a pre-selected set of ontologies, like tools/approaches presented in [168, 127, 151, 89].

In an open IoT ecosystem in which data is not modeled to suit a single IoT platform, but instead based on common, community-based ontologies that could be understood by many platforms, ORTs are essential. The ORTs surveyed in this chapter could be used for this purpose, however, the variety of tools and that the recommendation differs based on the chosen tool (due to different ontology collections and selection capabilities) can be frustrating for users. Further, the success of ontology usage in the biomedical domain indicates that domain-dependent, administered ORTs have a higher chance of being used and adopted by the community and to achieve a consensus. An ORT specialized on IoT-related domains could provide unique collection and selection features, e.g., taking into account the number of IoT platforms and their capabilities that comply with a certain ontology.

4.4.3 Research Challenges and Directions for ORTs

The following discussion on challenges for ontology discovery is based on the dimensions of ORTs as previously introduced (cf. Figure 4.2 and Figure 4.3).

[4] `https://www.industrialontologies.org/`
[5] `http://www.eclipse.org/vorto/`

Collection. The collection of ontologies for IoT domains is a challenging task because most ontologies are still being proposed in the scope of research projects. LOV4IoT [84], e.g., is dedicated to classify proposed ontologies and to make them accessible by integrating them into the LOV platform [232]. One challenge for ontology collections of IoT domains is the restriction to certain domains, as sensors are deployed in an increasing number of settings (e.g., in cities and factories) and thus in new contexts that are required to be modeled. However, projects such as LOV4IoT indicate that ontology collections can be eventually maintained domain-independently. Future efforts to collect ontologies for IoT domains will help recommendation tools to build a better ontology candidate set.

Evaluation. The evaluation of IoT ontologies as such can rely on the quality criteria identified in the survey. However, since many ontologies are still being proposed for the same domain and due to the rapid pace of developments in the IoT, ontologies are being continuously improved. Hence, more emphasis can be put on the evolution of ontologies, i.e., focusing on ontologies that are being actively maintained and extended, and, on the other hand, filtering those that become out-of-date. The most critical qualitative evaluation of an ontology, its domain accuracy, requires human judgment. Future ontology evaluation tools, designed as collaborative platforms integrated in ORTs, will help to achieve a community consensus about proposed IoT ontologies.

Curation. The number of ontologies available on the one hand, and the complex task of reviewing ontologies on the other hand, call for semi-automated curation processes. A particular challenge for the IoT is to keep track of the developments, classify, and collect metadata of proposed ontologies that can be of interest to users and provide valuable information for matching and ranking ontologies. Despite its importance, a trend was identified in which recent tools rather focus only on curation or on ranking of ontologies. The combination of both and the provisioning of curated data to ranking models will benefit future recommendation tools. Moreover, the importance of matching of existing, well-known ontologies to achieve interoperability has been highlighted in the survey. A collaborative curation platform could support the ontology matching process and could serve as a documentation of the achieved matches, which can be taken into account when recommending an ontology based on a query.

Interaction. One challenge is the requirement for more expressive ways to formulate the information need of the different IoT ecosystem actors. This could, for example, correspond to query formats based on outputs from IoT gateways with proprietary data models (i.e., text corpora such as JSON, XML, etc.). and the specification of the intended use, such as data stream annotation, knowledge specification for a context-aware system, and IoT data discovery. Such improvements of tool's interfaces will foster the adoption of ontology recommendation by users and developers in IoT settings.

Matching. As highlighted previously, it is not a trivial task to decide whether a recommendation should be made on an ontology, term/entity level or based on combinations from different ontologies. This may not only depend on the interaction mechanisms provided but also on the intentions of the user. The development of more sophisticated matches with different levels of granularity and considering the user's intent, such as its IoT use cases, will help to optimize the recommendation task.

Ranking. The survey revealed that the popularity of ontologies/terms is a desirable feature for ontology recommendation. However, in the surveyed tools, this feature is computed only by analyzing datasets from the LOD cloud, which do not represent semantically annotated IoT data. This may result in miscalculated qualitative scores for IoT ontologies and does not provide an objectively appropriate ranking. One challenge is thus to define a popularity measure that is suitable for IoT ontologies. Possible directions include the employment of modern information retrieval techniques, such as analyzing the user click behavior of existing ORTs (that contain IoT ontologies and are used by IoT actors) to calculate the popularity of ontologies and terms. Lastly, existing ORTs do not consider more complex features for advanced users of ontologies, such as the reasoning complexity of an ontology [21, 244]. Understanding how the ontology recommendation influences reasoning capabilities and constraints in IoT applications is not trivial and opens new challenges. Specialized ranking models for IoT use cases, e.g., through additional information generated during the curation process, will significantly improve the overall recommendation and foster further convergence to most suitable ontologies for IoT use cases.

4.5 Conclusion

In this survey, the process of ontology recommendation was thoroughly reviewed and placed into the context of IoT ecosystems. ORTs help to guide actors of IoT ecosystems when publishing, discovering and integrating IoT data and services from heterogeneous sources. A comprehensive taxonomy was defined based on dimensions regarding the discovery (i.e., collection, evaluation, and curation) and selection (i.e., interaction, matching, and ranking) of appropriate ontologies. This framework served to evaluate 40 ORT from the literature and trends/findings with regard to the identified features were highlighted. Moreover, the conceptual integration of ontology recommendation in IoT ecosystem use cases and the consideration of ORTs in today's landscape of IoT platforms were discussed.

In conclusion, two dimensions of ontology recommendation are important: curating an ontology collection and providing simple, yet efficient selection mechanisms. The survey revealed that tools often focus on either one, and that implemented strategies for both differ greatly. It is not completely clear, however, how different features impact the overall recommendation quality. Even though first advancements of sharing and reusing data models defined by the community for the IoT could be evidenced, today's scope of IoT platforms does not yet consider ORTs. Whereas ORTs have been integrated in tools that support traditional Semantic Web use cases, only a few tools supporting use cases of IoT ecosystems with ontology recommendation could be found.

The presented framework is limited to functional requirements that impact the output of ontology recommendation. Non-functional requirements (e.g., performance, reliability, scalability) impose additional challenges (e.g., efficient indexing of ontologies) on the implementation of ORTs which have not been considered in the scope of this survey. The evaluation presented in this survey can support Semantic Web developers and IoT researchers in getting an overview of the state-of-the-art in ontology recommendation, and help to choose the most appropriate tool. Furthermore, the presented taxonomy can be used to compare newly proposed approaches to improve ontology recommendation with previous work.

We claim that a tool that serves as a platform to share, extend, curate, and recommend ontologies of IoT-related domains could serve as a fundamental building block for the convergence to interoperable IoT ecosystems. In the following chapters, we address several of the identified shortcomings of ORTs, focusing on the key recommendation dimension: *ranking*.

Chapter 5

Efficient Popularity-driven Ranking For IoT Ontology Collections

This chapter investigates challenges of conventional ontology ranking models for IoT ontology collections (RQ 2) and investigates the ranking effectiveness of qualitative ranking features (RQ 3). In particular, this study highlights issues that emerge from the evaluation criteria *popularity*, which in its common definition is ineffective for IoT ontology collections. Instead, we derive popularity-driven relevance labels from scientific publications of ontologies from the LOV4IoT ontology collection. We employ a LTR workflow and compare ranking models that rely on qualitative features that do not require any external metadata, but can be directly extracted from the ontologies. The study is motivated based on the review of the previous chapter, which highlights the circumstances that the impact of ranking features and especially in the context of IoT ontology collections is fairly unexplored. This chapter is based on the work that has been presented in the following paper:

- Niklas Kolbe, Sylvain Kubler, and Yves Le Traon. "Popularity-driven Ontology Ranking using Qualitative Features". In: *International Semantic Web Conference*. Springer. 2019. DOI: `10.1007/978-3-030-30793-6_19`

Contents

5.1 Introduction

In the Semantic Web, efficient ontology reuse is a key factor to enable and enhance the interoperability of computing systems [204]. Approaches to ontology ranking are a key component in finding and selecting the most relevant ontologies based on a query [193]. The importance of ontology reuse and recommendation is also increasing in IoT environments that aim for the seamless discovery, access and integration of heterogeneous, sensor-originated data through the Web. This chapter concerns limitations in the ranking effectiveness of state-of-the-art ontology ranking models for IoT ontology collections.

Motivation. This work is motivated by the need of researchers and practitioners to discover and select the most relevant ontologies for their needs. The large number of available ontologies and the fast-paced developments in domains often make it difficult to find and select the most appropriate ontologies. For the IoT case, this is evidenced through extensive surveys in the literature [70, 126, 88, 7]. This does not only concern ontologies with regard to sensors and sensor network setups, but further to sensor observations [88] (e.g., in the context of smart city use cases with regard to the environment, transportation, health, homes, and factories). At the core of many state-of-the-art tools that facilitate ontology reuse – such as repositories, search engines and recommender systems – lies the ranking of ontologies for a user query in the form of keywords.

Importance of popularity. Fundamental ontology reuse strategies rely on ontologies' *popularity*, which is typically understood as the measure of how often an ontology is used to model data in the LOD cloud [196]. While rankings foremost take into account the semantic match of query and ontologies in the collection, current state-of-the-art tools such as LOV [232], TermPicker [197], and vocab.cc [208] further incorporate such a popularity measure in their ranking model. This is crucial because it reflects the community's consensus on ontologies' relevance, instead of solely relying on how well ontologies semantically match the query. Thus, the approach of computing the popularity measure has an important influence on the performance of the ranking model.

Problem statement. We find that the approach to derive popularity from datasets of the LOD cloud, as computed in many state-of-the-art tools, can be problematic for ontologies of some domains. We illustrate this problem in Figure 5.1, which shows the number of ontologies contained in the well-known LOV platform that have never been reused in LOD datasets[1]. In total, only $\sim 35\%$ of the ontologies in the repository have been reused. We identify particular critical domains with no reuse in any LOD dataset for any ontology in the collection, namely: Services, Industry, IoT, Transport, and Health. We consider all these domains highly relevant to IoT application domains (e.g., smart mobility, smart health care, industry 4.0), which thus forms our motivating case to investigate qualitative ontology ranking from this perspective. From a more general viewpoint, this case highlights the problem that the likeliness of missing relevant information to explicitly determine popularity for all ontologies in a collection is high, leading to the computation of ineffective popularity scores in the ranking model.

Contributions. This research contributes to the extension of scope and effectiveness of popularity-driven ontology ranking models, aiming to make these models less

[1] Extracted from the LOD SPARQL endpoint: `https://lov.linkeddata.es/dataset/lov/sparql`

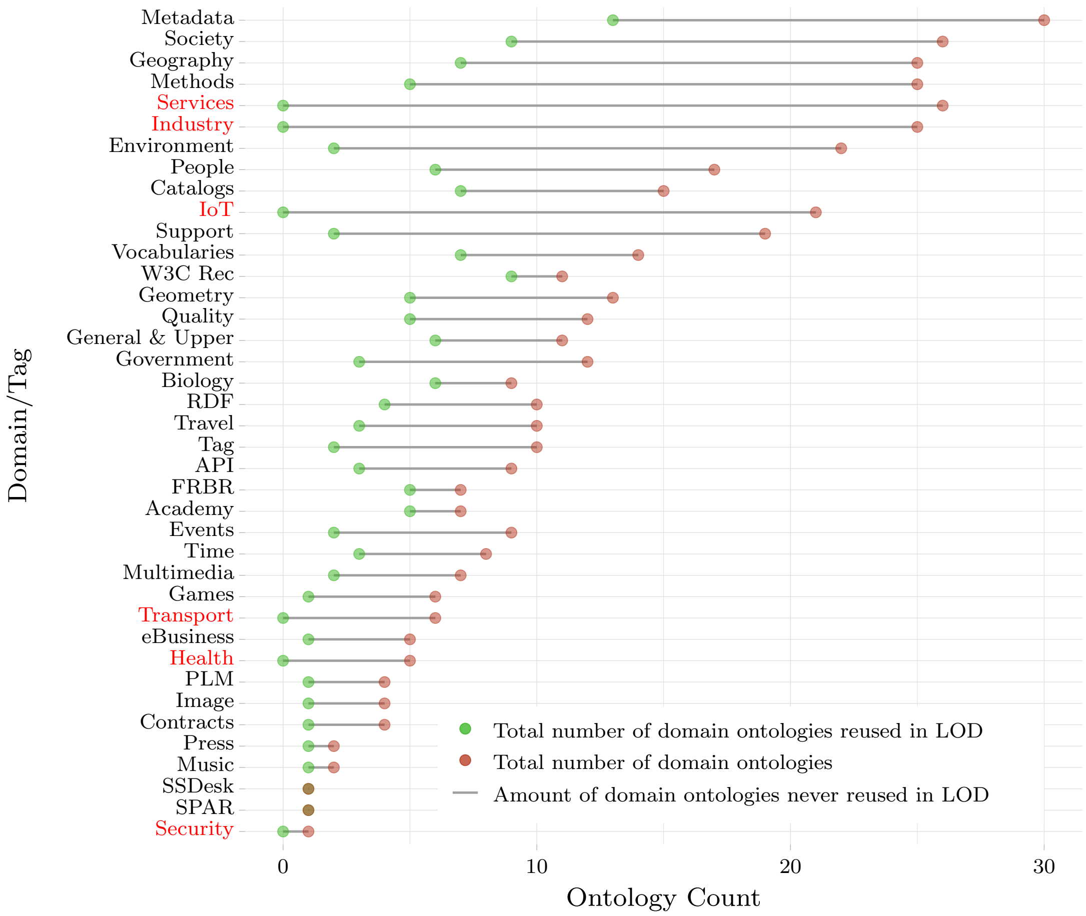


Figure 5.1: Count of ontologies per category in the LOV repository that were never reused in LOD datasets, which is often used as underlying popularity measure in state-of-the-art rankings. It shows that this score is inefficient for many domains related to WoT applications, for which none of the ontologies appear in any LOD dataset.

dependent on the underlying popularity measure, such as the selection of LOD datasets (and the way these datasets are assembled). In this respect, we investigate whether the relevance degree in terms of popularity can be predicted with qualitative properties of the ontology instead of relying on an explicit popularity feature as it is common in the state-of-the-art. We perform this study (based on the problematic IoT case) by learning a ranking model that uses the popularity as relevance degree for the prediction target. This approach to ontology ranking results in fairer scores for ontologies that were developed for use cases other than LOD publication, such as semantic sensor data annotation and the development of context-aware applications. In general, obtaining relevance labels for LTR is perceived as a major challenge and a costly process [132]. We propose a popularity measure for ontologies of IoT domains that relies on scholarly data (i.e., the citation history of ontologies' associated scientific publication) to determine relevance degrees in terms of popularity. This approach overcomes limitations of existing approaches, and we ensure that this measure approximates popularity in terms of reuses by evaluating the model on state-of-the-art rankings.

The remainder of this chapter is structured as follows. Section 5.2 defines the key ranking features and introduces the approach to relevance mining from scholarly data. The experiments, data collection and results are presented in Section 5.3. The findings

Table 5.1: Overview of Selected Ranking Features

Category	Feature		Description
Relevance	Φ_1	Lucene	A Lucene match with property boost.
	Φ_2	Word2Vec	Score based on closely related words of the query.
	Φ_3	WordNet	Score based on senses and synonyms of the query.
Importance	Φ_4	Availability	Whether the ontology is accessible at its URI.
	Φ_5	Believability	Whether provenance information is provided.
	Φ_6	Understandability	To which degree terms are labelled and commented.
	Φ_7	Interlinking	To which degree the ontology refers to external terms.
	Φ_8	PageRank	The importance derived through `owl:imports` statements.
	Φ_9	Consistency	Whether a reasoner does not detect inconsistencies.
	Φ_{10}	Richness (Width)	The size of the ontology in terms of width.
	Φ_{11}	Richness (Depth)	The size of the ontology in terms of depth.

are further discussed in Section 5.4; the conclusion follows.

5.2 Ranking Features and Relevance Mining

This section presents the selected ranking features that are considered to constitute our proposed model as well as our approach to derive relevance labels for ontologies of IoT application domains. The selection of ranking features is based on comprehensive studies in the literature on ontology ranking and quality [121, 247]. We include all attributes identified in survey [121] except for subjective features and those that only concern term ranking, not ontology ranking. Table 5.1 provides an overview of the selected features. Our interpretation of these features, as presented in the following, is guided by the review presented in [247].

5.2.1 Relevance Features

Relevance features aim to determine most suitable matches for a query and an ontology corpus, for which the following features are selected:

Lucene match (Φ_1). Our fundamental feature to find relevant ontologies based on keywords is a Lucene match [145]. As argued in [232], ontologies are structured documents and more meaningful matches should be given a higher score. We adopt the approach of [232] and apply a property boost to the Lucene match that aims at rewarding more important matches, such as local names, primary labels (e.g., `rdfs:label`), and secondary labels (e.g., `rdfs:comment`). The definition of the Lucene score is given in Eq. 5.1.

$$\begin{aligned}\text{Lucene}(Q,O,R) = {} & \text{coord}(Q,O)\cdot\text{queryNorm}(Q)\cdot \\ & \sum_{i=1}^{n}\big(\text{TF}(q_i,O,R)\cdot\text{IDF}(q_i,R)^2\cdot\text{propertyBoost}(q_i,R)\big)\end{aligned} \tag{5.1}$$

Word2Vec (Φ_2). Word2Vec [146] trains a neural network to predict the surroundings of a word. We employ this approach to find closely related words of the input search terms and compute a score based on the cosine distance and the Lucene match. The respective matching score is given in Eq. 5.2.

$$\text{Word2VecMatch}(Q,O,R) = \sum_{w\in\text{cosineDistance}(Q,M)}\text{cosineDistance}(Q,w_i,M_{w2v})\cdot\text{Lucene}(w_i,O,R) \tag{5.2}$$

WordNet (Φ_3). WordNet [148] is a lexical database in English. We use this source to find senses and synonyms of the keyword input and compute a score for these words based on the Lucene search, as given in Eq. 5.3.

$$\text{WordNetMatch}(Q,O,R) = \sum_{\substack{w_i \in \text{sense}(Q, D_{\text{WordNet}}) \,\cup \\ w_i \in \text{synonym}(Q, D_{\text{WordNet}})}} \text{Lucene}(w_i, O, R) \tag{5.3}$$

5.2.2 Importance Features

Importance features aim to assign a score to an ontology within a collection independently from the query. The selected features that represent the ontologies' quality are defined as follows:

Availability (Φ_4). The availability indicates whether ontology O can be accessed at its indicated URI. We derive this feature as given in Eq. 5.4.

$$\text{Availability}(O) = \begin{cases} 1, & \text{if httpResponseCode}(\text{URI}(O)) = 200 \\ 0, & \text{otherwise} \end{cases} \tag{5.4}$$

Believability (Φ_5). The believability of a published ontology increases with the presence of provenance data (e.g., specification of authors and descriptions), and is computed based on Dublin Core Metadata Initiative (DCMI) metadata terms[2], as given in Eq. 5.5.

$$\text{Believability}(O) = \begin{cases} 1, & \text{if } \{\text{URI(O) } \texttt{dcterms:creator ?c}\} \cup \\ & \quad \{\text{URI(O) } \texttt{dcterms:description ?d}\} \neq \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

Understandability (Φ_6). The better an ontology is documented, the easier it is to reuse it. We measure the understandability of an ontology by computing how many of all defined terms in ontology O are labeled and commented.

$$\text{Understandability}(O) = \frac{|\,\text{labelledTerms}(O)|}{|\,\text{definedTerms}(O)|} + \frac{|\,\text{commentedTerms}(O)|}{|\,\text{definedTerms}(O)|} \tag{5.6}$$

Interlinking (Φ_7). Ontologies foster interoperability by establishing links to previously defined terms. Thus, we count the out-links found in an ontology as formalized in Eq. 5.7.

$$\text{Interlinking}(O) = |\,\text{outlinks}(O)| \tag{5.7}$$

PageRank (Φ_8). PageRank [164] is an algorithm that helps to compute the importance of ontologies based on how often they have been referred to by others (i.e., in-links). We compute the PageRank score based on `owl:imports` statements, as given in Eq. 5.8.

$$\text{PageRank}(O_i, R) = \frac{1-d}{|R|} + \sum_{O_j \epsilon \, \text{importedBy}(O_i)} \frac{\text{PageRank}(O_j, R)}{|\,\text{imports}(O_j)|} \tag{5.8}$$

Consistency (Φ_9). Ontologies are expected to be logically consistent, which can be derived through OWL reasoners. We compute the consistency feature as given in Eq. 5.9.

$$\text{Consistency}(O) = \begin{cases} 1, & \text{if } \{\text{inconsistencies}(O)\} = \varnothing \\ 0, & \text{otherwise} \end{cases} \tag{5.9}$$

[2] `http://purl.org/dc/terms/`

Richness (Φ_{10} & Φ_{11}). We further consider the size of the ontology in the form of its width (see Eq. 5.10) and depth (see Eq. 5.11).

$$\text{Width}(O) = |\,\text{typeStatements}(O)| \tag{5.10}$$

$$\text{Depth}(O) = |\,\text{subClassOfStatements}(O)| + |\,\text{subPropertyOfStatements}(O)| \tag{5.11}$$

5.2.3 Relevance Mining Approach

The employed LTR is a supervised machine learning approach that requires relevance labels for query-ontology pairs. We propose to derive a popularity measure based on corresponding scientific publications associated with an ontology. We are inspired to follow this approach as a large number of ontologies for IoT application domains emerge from research projects, as evidenced in [70, 126, 88, 7]. Furthermore, it overcomes several limitations of other approaches: (i) as previously discussed, LOD does not provide a reliable source for ontology reuse in IoT application domains; (ii) deriving relevance through user click logs requires access to closed back-ends of existing ontology search engines with a large user base; (iii) human labeling is costly and, unlike mining relevance from scholarly data, does not come with the benefit of being reproducible.

Our popularity score is based on two measures; (i) $\mathit{citationsPerYear}(O)$: citations per year are counted and divided by the number of ontologies described in the same publication to represent the overall impact of the ontology; and (ii) the $\mathit{linearTrend}(O)$: a linear regression of the citation history to reward positively trending ontologies combining the intercept and the slope of the linear model. The final relevance score, as given in Eq. 5.12, is the mean of both min-max normalized measures and used to derive the total order π_l for the set of ontologies associated with a query, for which an ontology with a higher popularity score is more relevant than another, i.e., $Rel_a \succ Rel_b$ if $\mathit{popularity}(O_a) > \mathit{popularity}(O_b)$.

$$\text{popularity}(O) = \frac{\text{citationsPerYear}(O) + \text{linearTrend}(O)}{2} \tag{5.12}$$

A ground truth mining process is always assumed to contain bias and noise: for relevance mining from scholarly data, all self-citations are subtracted from the citation history, and incomplete years are not considered (i.e., citations of the current year and of the year of publication). Although the citation history is often used to measure a study's impact, the associated reason for the citation remains unknown, which is a potential threat to the validity of our popularity scores. We assume that the proposed measure reflects the overall ontologies' relevance for the scientific community (e.g., we assume that for outdated ontologies, the citation count will decline and the score is penalized accordingly through the linear trend). In the following experiments, the proposed ranking model is tested on completely independent datasets to evaluate whether our training data is accurate and the assumptions hold.

5.3 Experiments

This section presents the experiments following the LTR approach to build a ranking model with qualitative properties of the ontologies to predict the relevance degree. An overview of the following experiments is illustrated in Figure 5.2, whose aims are twofold; (i) to investigate whether qualitative features in the ranking model help to improve the ranking performance with regard to the relevance degree, and (ii) to confirm

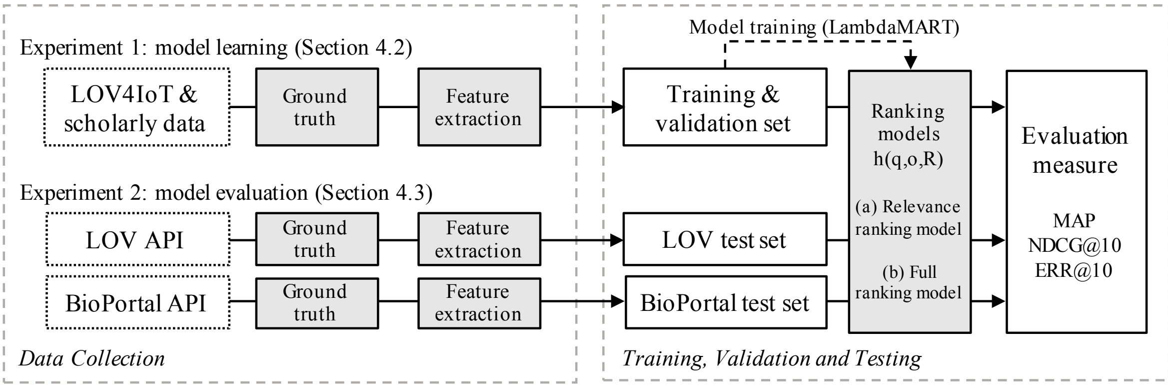


Figure 5.2: Experiment overview.

the validity of the results by testing the model on data sets derived from state-of-the-art ontology rankings.

5.3.1 Experiment Design

The design choices to learn and evaluate the ranking model are as follows:

Learning algorithm: various LTR algorithms were proposed by the machine learning community. The ranking model is trained using the list-wise LambdaMART algorithm which has successfully been applied for real-world ranking problems [33] and has also been previously selected in related work for ontology ranking [37]. We rely on the LambdaMART implementation of the RankLib[3] library.

Evaluation metrics: the performance of the ranking model is validated and tested based on the Mean Average Precision (MAP) [132], Normalized Discounted Cumulative Gain (NDCG) [132] and the Expected Reciprocal Rank (ERR) [43], considering the first ten elements (k=10). A unified point-wise scale for relevance labels is required for some evaluation metrics, so popularity scores of query-ontology pairs are mapped to a scale of 0-4 for the experiments. While MAP is only a binary measure (i.e., 0: considered not relevant, 1-4: considered equally relevant), the NDCG and ERR scores do consider the multi-valued relevance labels (i.e., these metrics consider how well the ranking model matches the relevance degree 0-4). Whereas NDCG only depends on the position in the ranking, ERR discounts the results appearing after relevant ones, which supposedly better reflects the user behavior of search engines [43]. The ranking model is trained by optimizing the ERR@10 score using 10-fold cross-validation, meaning that the training data is randomly partitioned into ten equal-sized subsamples. Iteratively, nine of these folds are used for training and the remaining one for validation.

Feature sets: the training dataset is prepared by extracting the feature vectors for each query-ontology pair as introduced in Section 5.2. We rely on the Lucene search engine of the Stardog[4] triple store, the openllet[5] OWL reasoner to infer consistency and the GloVe word vector model[6] to compute the Word2Vec feature.

[3]`https://sourceforge.net/p/lemur/wiki/RankLib/`
[4]`https://www.stardog.com/`
[5]`https://github.com/Galigator/openllet`
[6]`https://github.com/stanfordnlp/GloVe`

5.3.2 Ranking Model Training and Validation

In the first experiment, we train and validate the ranking model, as presented in the following.

Data collection: the data for training and validation is collected from the LOV4IoT catalog[7]. 455 ontology files related to IoT applications could be downloaded through the catalog (each file being treated as a separate ontology). Only 433 files were syntactically correct and stored as named graphs in a local triple store. We derive training examples by using the available classification labels from the LOV4IoT catalog as queries (i.e., ontologies' domain[8] and described sensor devices[9]), and consider the correspondingly tagged ontologies as relevant. As previously motivated, we rely on scholarly data to derive degrees of relevance. From the initial collection, 395 ontologies could be assigned to 125 different scientific publications based on the LOV4IoT metadata. This collection resulted in $1.1M$ triples with $133K$ distinct terms and formed the ontology repository for the experiments. The citation history from Google Scholar of the assigned publications is used to derive a relevance score for the ontologies based on the approach presented in Section 5.2. The resulting scores are mapped to relevance labels 1-4 by dividing the range of the highest and lowest popularity score for each query into four equal-sized intervals, and a random set of irrelevant ontologies is added with the relevance label 0. The resulting ground truth contains 1028 query-ontology relevance judgments with 25 different queries, for which the previously introduced ranking features are extracted to finalize the training set.

Experiment and results: the first experiment aims at investigating whether the selected qualitative importance features improve the ranking performance with regard to the relevance degree. Thus, we first train and validate a model only based on relevance features, and use this as a baseline to evaluate the performance of a model that further considers the importance features. The results are summarized in Figure 5.3, showing the performance of two ranking models: the relevance model (a) is only trained with the relevance features (Φ_1 - Φ_3), whereas the full model (b) also includes the importance features ($_1$ - Φ_{11}).

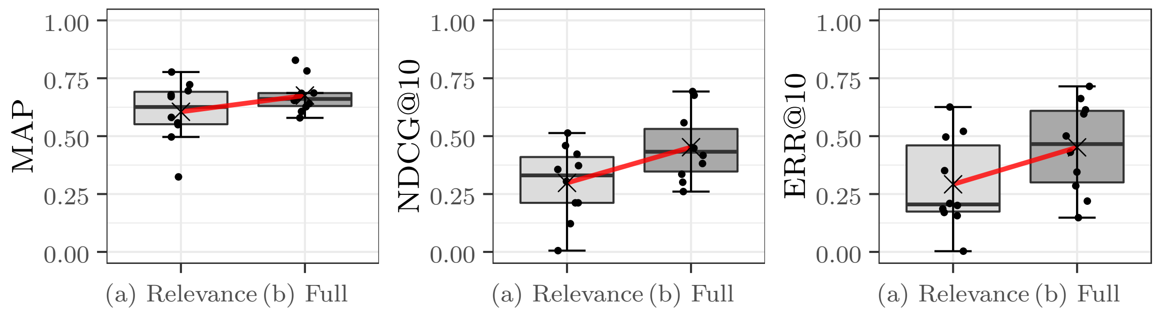


Figure 5.3: Comparison of trained models with regard to MAP, NDCG@10 and ERR@10 on the validation set, for model (a) using only relevance features (Φ_1 - Φ_3) and model (b) using further the importance features (Φ_1 - Φ_{11}). The red lines indicate the difference of the respective metric's mean between the two models.

The results show that the trained ranking models appear to appropriately rank ontologies with regard to their relevance. We observe that the addition of qualita-

[7] http://lov4iot.appspot.com/
[8] Denoted by `m3:hasContext`
[9] Denoted by `m3:hasM2MDevice`

tive features only has a small impact on the MAP score, but significantly improves the NDCG@10 and ERR@10 scores. This behavior is expected, as MAP effectively only measures the semantic match of query and relevant ontologies, whereas the qualitative features aim at ranking relevant ontologies according to their relevance degree. NDCG@10 and ERR@10 both reflect this degree, as they take into account multi-valued relevance labels. We thus conclude that qualitative features helped to improve the ranking with regard to the popularity-based relevance degree captured in the ground truth. Subsequently, this implies that the proposed approach can extend the scope of state-of-the-art rankings, by improving the performance for domains in which ontologies were never reused in LOD datasets. In such cases, the explicit popularity feature always results in the same score for all ontologies (i.e., zero) and effective ranking is only based on relevance (i.e., corresponding to model (a)). The presented approach, in contrast, predicts the popularity based on the qualitative features (i.e., corresponding to model (b)), even when no explicit information of popularity or reuse is present.

5.3.3 Ranking Model Evaluation and Comparison

The second experiment aims at evaluating and comparing the model with independent datasets derived from state-of-the-art rankings. We do this in order to ensure that our assumptions for the ground truth, as introduced in Section 5.2, hold and to confirm whether the findings from the first experiments are valid. Due to the lack of existing benchmarks and implementations of ranking models proposed in the literature, we derive test sets from state-of-the-art tools which must: (i) provide an open API that returns the computed ranking score of the top-ranked ontologies for a query; (ii) make the underlying ontology collection available for download; and (iii) incorporate a popularity measure in their ranking model. We choose to compare the proposed ranking model to approaches from two different domains that fulfill these requirements: the LOV repository [232], which measures popularity based on LOD occurrences (by excluding the problematic domains without any reuse in LOD for the test sets); and the NCBO 2.0 of the BioPortal [141], which ranks biomedical ontologies and covers ontology's popularity in its notion of acceptance, derived by the number of other curated repositories that also keep an ontology in its collection.

Data collection: we create the test sets based on the LOV REST API[10] and the BioPortal REST API[11]. For each platform, we (i) derive a set of test queries by extracting nouns and verbs from names and descriptions of all ontologies in the respective repository, (ii) use each test query to retrieve the ranking from the respective API that forms the ground truth, (iii) use the same strategy as for the training data to map the ranking scores to a scale of 1-4 and add a random set of irrelevant ontologies with a relevance of 0, and, lastly, (iv) complete the test set by extracting the features for all query-ontology pairs from a local triple store that contains the respective ontology collection. For the LOV test set we only consider domains with at least five ontologies that have been reused in LOD datasets, in order to ensure that the derived ground truth sufficiently reflects the ontologies' popularity (see Figure 5.1). This process resulted in test datasets with 2998 (LOV) and 4313 (BioPortal) query-ontology relevance scores.

Experiment and results: in the second experiment, we test both the validated

[10] `https://lov.linkeddata.es/dataset/lov/api`
[11] `http://data.bioontology.org/documentation`

relevance model (a) and the full model (b) from the first experiment on the newly derived datasets. The results are illustrated in Figure 5.4, showing the comparison of the performance for the LOV and BioPortal test set, as well as the mean performance of the full ranking model from the first experiment (indicated by the dashed lines).

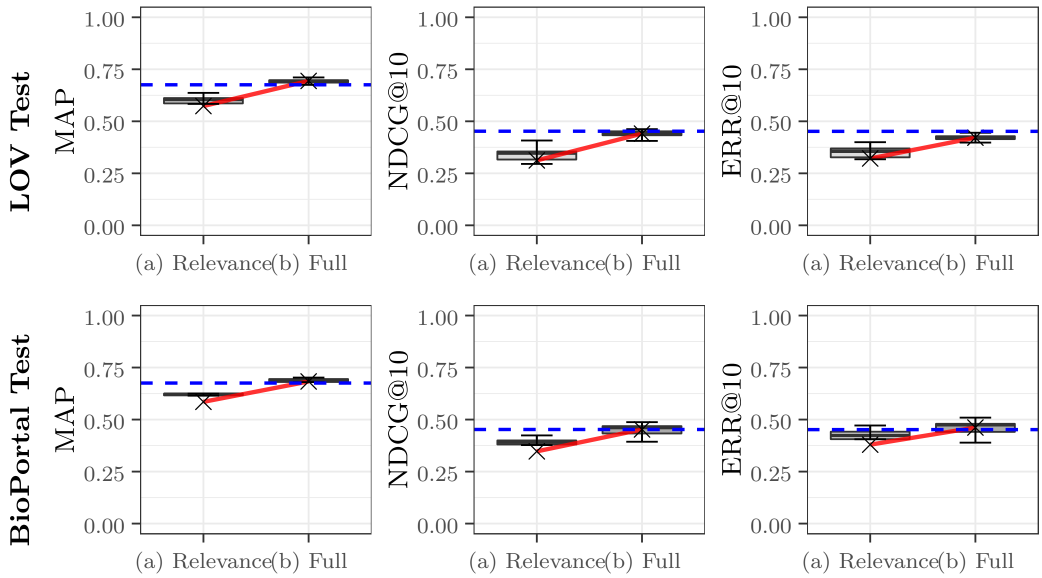


Figure 5.4: Comparison of the validated ranking models from the first experiment with the LOV and BioPortal rankings. The dashed lines indicate the mean performance of the full model on the 10 fold validation sets, showing that the model performs similarly well on the test datasets. The red lines indicate the difference of the respective metric's mean between the two models.

The experiment results lead to two important conclusions. First, it shows that the learned models behave reasonably well on these completely independent datasets, evidenced by a similar performance compared to the first experiment. This confirms that the underlying ground truth to train our model is valid and, subsequently, implies that the citation history of ontologies in IoT domains is a fair approximation of their popularity. Secondly, we observe a similar behavior of the relevance and the full ranking model as in the first experiment, for which the full model improves the ranking in terms of relevance degree. Albeit the improvement on test sets is lower as in the previous experiment, it shows the same trend and thus validates our previous conclusion that the selected qualitative features help to predict the popularity-driven relevance degree of ontologies. The experimental results are further analyzed and discussed in the following.

5.4 Discussion

Experiment summary. This study reveals that the prediction of ontologies' relevance for a query in terms of popularity can be improved with qualitative features. This confirms the hypothesis of a correlation between ontologies' popularity and its quality, based on the intuition that ontologies with better quality are more likely to be reused than others of the same domain. The presented approach extends the scope and applicability of the ranking model, as it is not dependent on measures of LOD occurrences. As motivated previously, this approach gives a fairer score to ontologies that are not engineered for LOD publication purposes, such as IoT application domains,

Table 5.2: Full Model Feature Frequencies Averaged Over All Folds

Category	Feature		Avg. Freq.
Relevance	Φ_1	Lucene	1056.9
	Φ_2	Word2Vec	680.0
	Φ_3	WordNet	1375.5
Importance	Φ_4	Availability	697.7
	Φ_5	Believability	55.8
	Φ_6	Understandability	1237.9
	Φ_7	Interlinking	634.8
	Φ_8	PageRank	1302.1
	Φ_9	Consistency	777.7
	Φ_{10}	Richness (Width)	535.1
	Φ_{11}	Richness (Depth)	646.5

and furthermore also for newly proposed ontologies without any reuses that are well-defined.

Influence of qualitative attributes. The LambdaMART algorithm applied in the experiments creates an ensemble of regression trees that can be further analyzed to better understand the model and its consequences. One way to infer the importance of each feature on the ranking model is the frequency it was used for classification of the training examples. We use these counts to discuss the model's implications and directions for future research. Table 5.2 reports the results for feature frequency. We derive the following insights based on the feature statistics, albeit detailed experimentation would be required to confirm them. One interesting observation is that the feature believability (Φ_5) barely contributes to the model and would be the first candidate to be replaced with another feature. This is surprising, as other approaches fundamentally rely on provenance information such as ontologies' authorship to compute the ranking [210]. Other observations include that an ontology's incoming links (Φ_8) appear to have much more significance than outgoing links (Φ_7). This is intuitive, as being imported by another ontology often requires the ontology to be considered relevant by ontology engineers other than the original authors. In addition, it can be observed that features that solely reflect the internal graph structure (Φ_{10} and Φ_{11}) are less often used by the model than more expressive qualitative scores such as understandability (Φ_6), consistency (Φ_9) and availability (Φ_4).

Implications of the proposed ranking approach. The experimental results of this study show that the proposed approach is promising to extend the scope of ontology ranking models. As evidenced through the experiments, this approach can also be adapted for other domains and we expect a model trained on domain-specific ontologies to perform better. This encourages further experimentation with more quality attributes, new interpretations of them, and with training sets from other domains in order to confirm the findings and achieve the development of better performing ranking models. The quality of LTR approaches also highly depends on the size of the training data. We expect future research to provide larger benchmarks that allow for the study of more complex models and better comparisons of ranking approaches, such as ground truths derived from user click logs of existing search engines. In a broader context, this approach to ranking could also encourage ontology engineers to put even more emphasis on qualitative traits of proposed ontologies in order to increase exposure and reuse in applications. Albeit the extraction of qualitative features can be computationally very expensive, these scores are independent of the user query and can be pre-computed. Thus, the lookup of these scores and re-ranking of relevant ontologies only has a minor impact on the run-time performance compared to the complexity of the semantic similarity search in the entire ontology corpus.

Novel ontology ranking model for the IoT. To the best of our knowledge, the proposed full ranking model is the first that effectively considers popularity for ontologies in IoT application domains. We thus conclude that the proposed full ranking model contributes to ontology selection for these domains in the scope of open IoT ecosystems, e.g., for ontology collections such as the LOV4IoT catalog. The ranking model can be integrated in more complex user interfaces and combined with various other selection criteria in IoT domains, that, e.g., further consider important standardization efforts.

Limitations. A potential threat to the validity of this study's experimental findings is the ground truth derived through popularity measures from scholarly data. While it is a common approach to use implicit user feedback as relevance score (such as user clicks), using citations arguably is a more ambiguous measure. Yet, as previously mentioned, this approach overcomes limitations of alternatives and our evaluation showed reasonable performance. We conclude that further experimentation is required in order to confirm whether similar observations can be made for other domains than WoT, by using training examples with a relevance score derived from other popularity measures. From an ontology reuse perspective, this study is limited as it only considers the ranking of single ontologies. However, practitioners often search for terms (e.g., as offered by LOV [232]) or combinations of ontologies (e.g., as offered by NCBO 2.0 [141]).

Resource availability. The derived datasets, source files to replicate the experiments, as well as more detailed results of the ranking models are available online[12], and may be used for future experiments and comparison studies.

5.5 Conclusion

In this chapter, we show that the prediction of ontologies' relevance in terms of popularity can be improved with qualitative features in the ranking model, making the model independent from explicit computed popularity metrics such as LOD occurrences. Moreover, we present a ranking model that effectively ranks ontologies of IoT domains with respect to their popularity. We show that the proposed model performs similarly well on test set derived from rankings of state-of-the-art tools, which is encouraging to adopt the presented approach also in other domains. Lastly, we discuss the importance of the qualitative features on the overall performance of the ranking model. The proposed model can be integrated in ontology selection mechanisms for practitioners and researchers in IoT use cases and thus contributes to establishing semantic interoperability in emerging large-scale IoT ecosystems. x As evidenced through this study, evaluation of ontology ranking models is a challenging task. In the next chapter, we address the challenge of ontology ranking evaluation as a general problem in the Semantic Web.

[12]Supplemental material: `https://tinyurl.com/y64sa6le`

Chapter 6

Ontology Ranking Benchmark

This chapter is concerned with the lack of sufficient benchmarks in order to compare different ontology ranking models as proposed in the literature (RQ 4). In order to develop a representative benchmark, implicit real-world user feedback of the LOV platform in the form of views and clicks is analyzed to derive a large dataset with relevance labels for ontology term search. This dataset, named LOVBench, is then used for an empirical comparison of ranking models as proposed in the literature to provide further domain-independent insights about their effectiveness (RQ 3). This chapter is based on the work that has been presented in the following paper:

- Niklas Kolbe, Pierre-Yves Vandenbussche, Sylvain Kubler, and Yves Le Traon. "LOV-Bench: Ontology Ranking Benchmark". In: *Proceedings of The Web Conference (WWW)*. ACM, 2020. DOI: 10.1145/3366423.3380245

Contents

6.1 Introduction

Ontology search provides users with a ranked list of either ontologies[1] or their terms for a given query. Efficient ontology search helps users to find and reuse existing knowledge on the Web [60, 115, 38, 95, 186], which benefits communities by establishing consensus on domain conceptualizations (e.g., as in the biomedical domain [206]) as well as the breakup of vertical data silos (e.g., as in Open Data [25] and the IoT [121]). This, ultimately, eases the discovery and reuse of Web data, fostering interoperability among computing systems.

However, the evaluation of ontology ranking models that were proposed in this context is relatively unexplored. The heterogeneity of evaluation strategies and underlying ontology collections makes it difficult to understand, interpret and compare the performance of proposed ranking models. Existing datasets for ontology ranking evaluation, such as the state-of-the-art benchmark CBRBench [38], are often built based on explicit judgments from human experts. However, the laborious approach to building such benchmarks led to relatively small datasets (in the case of CBRBench, only comprising ten queries with 819 relevance judgments). Given that ontology collections typically contain hundreds of ontologies and thousands of terms, benchmarks that only comprise ten queries are not sufficient for the evaluation of ranking models in real-world settings.

To overcome this issue and address the lack of ontology ranking comparison studies, we propose the ontology ranking benchmark *LOVBench*. It comprises both a dataset including a ground truth for ranking evaluations and an empirical comparison of state-of-the-art ranking models. The LOVBench dataset contains more than 180,000 inferred relevance judgments for more than 7,000 queries. LOVBench's ground truth relies on implicit, real-world user feedback in the form of queries and clicks that were collected from the LOV platform[2] [232], which comes with the following advantages. First, the relevance judgments for ontology terms to a query in LOVBench stem from feedback of many actual users, which is more representative than judgments from a few experts; second, collecting user feedback through search logs does not require manual effort, meaning the benchmark can be continuously updated to capture the evolution of ontologies and their relevance in the Semantic Web. The empirical evaluation is performed based on Learning To Rank (LTR) [132]. Compared to combining ranking features manually, LTR allows to learn the optimal combination of the considered features using supervised machine learning techniques and, thus, allows for a fair comparison of complex ranking models. We measure the models' ranking quality using the Normalized Discounted Cumulative Gain (NDCG) [111] as evaluation metric and we compare ranking configurations as proposed in DWRank [37], AKTiveRank [6] and CBRBench [38]. Our experiments confirm that all considered models outperform a straight-forward Lucene search [145]. We further show that the consideration of features that are well-suited to the user behavior as well as the consideration of much larger feature sets (given a large enough dataset like LOVBench), can significantly improve the ranking performance. The insights gained in this study help to understand the effectiveness of ontology ranking models and can serve as guidelines for the design of more efficient ontology ranking tools in terms of ranking quality.

In summary, the main contributions of this chapter are as follows:

[1]In the Semantic Web, ontologies are also referred to as *linked vocabularies*.

[2]`https://lov.linkeddata.es/dataset/lov/`

1. We provide insights of real-world user behavior in ontology search of which some dissent dominant assumptions in the literature;

2. we introduce and publish the LOVBench dataset, a large-scale ontology ranking benchmark based on actual and timely user feedback that allows for continuous updates and enables comparison studies using LTR;

3. we provide an experimental evaluation of ranking effectiveness of three state-of-the-art ontology ranking models (composed of 13 distinct ranking features) and propose feature variations and configurations that outperform them (using in total 33 distinct ranking features);

4. we conclude with recommendations for the design of efficient ontology search tools.

An overview of existing benchmarks and LOVBench is presented in Table 6.1. In contrast to the related work, LOVBench satisfies all the following: it relies on keyword-based queries, ranks classes as well as properties of ontologies from various domains, graded relevance labels are mined cost-efficiently from actual, real-world user feedback (views and clicks), and it provides sufficient judgments for complex models in LTR settings (7,395 queries and 184,224 relevance judgments). Moreover, our comparison of state-of-the-art ranking models considers best combinations of originally proposed along with newly designed features (in total 33), and we make the dataset as well as the feature implementation publicly available.

The remainder of this chapter is structured as follows. We present our approach to building an ontology ranking benchmark based on user clicks in Section 6.2. We evaluate the usefulness of user feedback collected through LOV in Section 6.3 and describe the details of the LOVBench dataset in Section 6.4. We report the subsequent empirical evaluation in Section 6.5 and discuss our findings in Section 6.6. Finally, we conclude the chapter in Section 6.7.

6.2 Approach

In this section, we present our approach to benchmarking ontology ranking models. As illustrated in Figure 6.1, our approach comprises four main steps: (1) analyzing the collected LOV search logs (containing users' queries and clicks), (2) inferring and evaluating relevance labels, (3) creating the LOVBench dataset, and (4) performing the empirical evaluation of ontology ranking models. In the following, we present an overview of each step and respective outcomes, including key features for the design of LOVBench.

Analyzing LOV search logs. In the first step (cf. Section 6.3.1), we analyze the LOV search logs to confirm whether the logs fulfill basic requirements to generalize user behavior for ontology search. Moreover, we extract fundamental insights concluded from the observed real-world user actions in the context of literature assumptions on ontology ranking design, which guide our later feature design and discussions on experimental results.

Inferring relevance labels. In the second step (cf. Section 6.3.2), we learn a user click model [48] based on the LOV search logs in order to infer relevance labels for query-term pairs. Considering user's reactions (i.e., clicks) on presented terms for a query to infer relevance labels has several advantages. Compared to obtaining explicit

Table 6.1: Comparison of Existing Benchmarks and the Proposed LOVBench Dataset

	AKTiveRank [6]	**CARRank [239]**	**CBRBench [38]**	**TermPicker [197]**	**IoT Ranking [120]**	**LOVBench**
Year	2006	2008	2014	2016	2019	2020
Query	Keyword	Keyword	Keyword	Triple-Pattern	Keyword	Keyword
Ranking	Ontology	Class	Class	Triple-Pattern	Ontology	Term
Source	Four experts	Four experts	Ten experts	Mining (LOD)	Mining (scholar)	Mining (logs)
Labels	List	List	Point (graded)	Point (binary)	Point (graded)	Point (graded)
Queries	7	80	10	5,650	25	7,395
Judgments	13	~ 400	819	3,010,620	1,028	184,224
Features	4	1	8	5	11	33
Dataset	No	No	Yes	Yes	Yes	Yes
Impl.	No	No	No	No	No	Yes

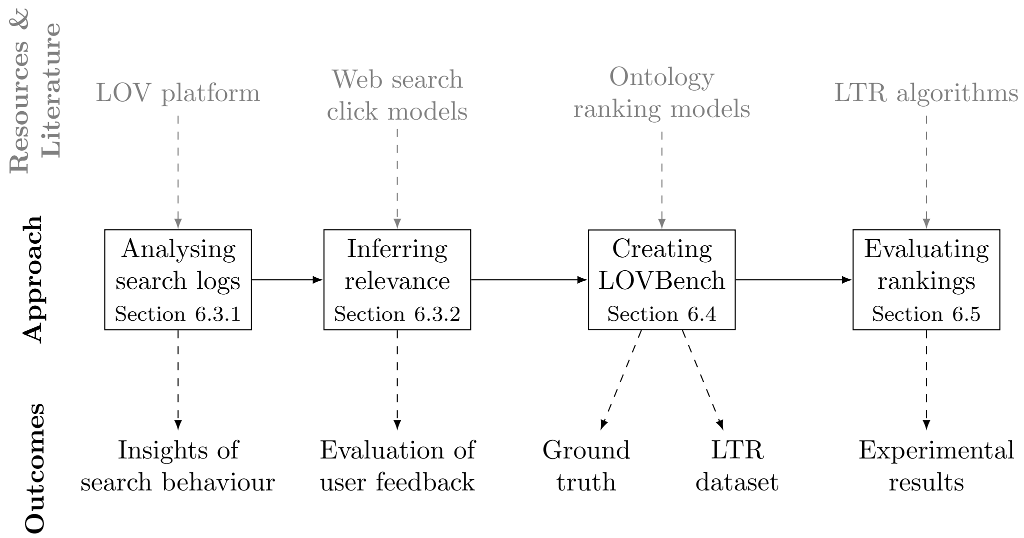


Figure 6.1: Overview of the LOVBench approach.

judgments from human experts, which may contain bias and become outdated, collecting implicit user feedback through logs is cost-efficient and further captures actual, potentially time-varying user preferences [236]. This helps to overcome the problem that the amount of training data in LTR settings is often low [4, 120]. However, clicks cannot be used as direct relevance judgments due to user's inherent biases (e.g., position bias), click incompleteness (missing feedback for relevant terms), and noise (a click does not necessarily imply relevance of a term) [2]. The purpose of user click models is to take these considerations into account and allow for the inference of actual relevance based on observed user clicks. In order to evaluate whether the LOV search logs allow us to infer meaningful relevance labels, we pursue the following steps:

1. We learn and compare several user click models with well-established assumptions of user behavior in Web search,
2. subsequently infer relevance judgments for query-term pairs contained in the logs, and
3. evaluate these relevance predictions by comparing them with judgments from human assessors of CBRBench [38].

These steps allow us to build a benchmark with relevance labels that are best aligned with both, observed user clicks as well as expert judgments.

Creating LOVBench. In general, an ontology benchmark is composed of two main components: first, a set of sampled query-term pairs with relevance judgments (the ground truth) and second, the extracted scores of selected ranking features for these query-term pairs (used as input for LTR experiments). In the third step of our LOVBench approach (cf. Section 6.4), we first sample terms for the ground truth based on common practices (similar to the one followed by LETOR [179]). Second, our strategy for the selection of ranking features for LOVBench is guided by related literature surveys [121, 60, 247], and respectively discovered features are selected based on applicability constraints. In summary, we build the LOVBench dataset as follows.

1. For each query contained in the search log, we sample relevant and non-relevant terms following common practices,

2. we infer the relevance labels based on the best-performing user click model of the previous step,

3. we select ranking features from the literature and propose variations, and

4. we extract corresponding feature scores for each query-term pair of the sample.

As a result, we derive a ground truth file for ontology evaluations and a dataset that can be used for LTR experiments.

Evaluating ranking models. The last step of our benchmark (cf. Section 6.5) is concerned with the comparison of the ranking models' performance based on the LOVBench dataset. We follow a standard LTR experimental methodology [132] and use standard evaluation metrics for information retrieval (NDCG). In addition to feature configurations as proposed in the literature, we propose novel variations that are motivated by the observed user behavior from the LOV search logs.

6.3 LOV User Feedback Evaluation

In this section, we provide insights of real-world user behavior for ontology search. We analyze the search logs in Section 6.3.1 and present the inference of relevance labels in Section 6.3.2.

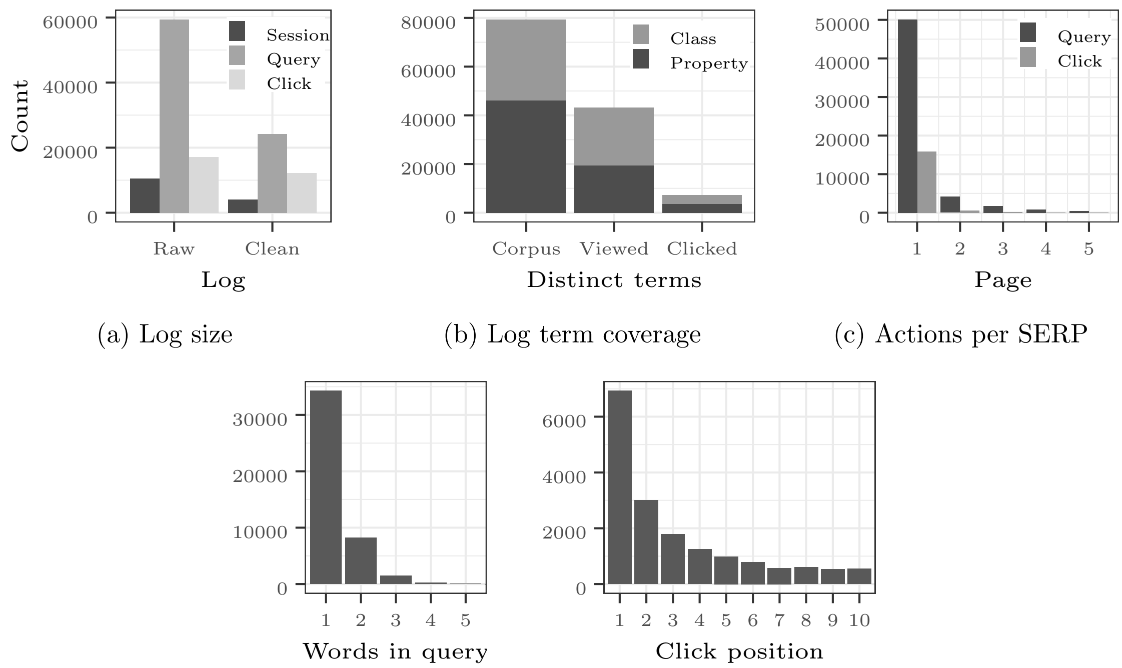
6.3.1 Search Log Analysis

The ontology collection of LOV, at the time of writing, consists of 680 ontologies associated with 43 different domains. User interactions with the LOV interface are logged upon queries and clicks, without storing any user information. Figure 6.2 provides details of the query and click log files collected from LOV's term search of a 7-months period starting from 01/2019.

The raw log files contain 10,579 user sessions with 59,398 queries and 17,125 clicks (see Figure 6.2a). In Figure 6.2b, we illustrate the amount of terms that are defined in the LOV corpus that have also been viewed and clicked by users. The coverage shows that the LOV search logs are sufficient to generalize search behavior with regard to the corpus size. The pre-processing for the clean logs include removing sessions without clicks, queries containing personal information, as well as queries beyond the first Search Engine Result Page (SERP). The latter is motivated by our observation that only very few users made use of the pagination feature (i.e., only few actions were logged beyond the first page), as shown in Figure 6.2c. We further made the following observations regarding the user behavior of which some are in contrast with assumptions made in the literature.

Importance of property ranking. The literature often focuses on class ranking, and some ranking and evaluation approaches do not consider properties [6, 38, 37]. As illustrated in Figure 6.2b, we observe equal amount of views and clicks of classes and properties in LOV, which should be considered when designing ranking features.

Multiple words in keyword query. Some approaches assume that user queries contain multiple words, e.g., when considering the number of words that match



(a) Log size (b) Log term coverage (c) Actions per SERP

(d) Query length (e) Clicks on first SERP

Figure 6.2: LOV search log analysis.

the term [37] and relying on the distances in the ontology graph of matched terms for each word in the query [6]. However, as shown in Figure 6.2d, we observe that the majority of queries in LOV are single words, which means the potential usefulness of these ranking models is limited to a small fraction of queries.

Position for evaluation. The frequency of click positions of the first SERP is illustrated in Figure 6.2e, which shows a strong bias towards the first position in the result list. Ranking evaluations should thus not only consider the complete SERP (first ten positions), which is a common choice in the literature [38, 37], but further based on metrics that only consider the more important top positions of the SERP.

These considerations guide our approach for the creation of LOVBench, as well as the design and discussion on the LTR experiments in later sections.

6.3.2 Relevance Inference

In this section, we learn several user click models from the cleaned LOV search logs in order to infer meaningful relevance labels from the implicit user feedback. We then evaluate their accuracy through a comparison with CBRBench, which contains judgments from human experts. We first describe the setup and subsequently present the evaluation results.

6.3.2.1 Learning User Click Models

In order to model user click behavior from the LOV search logs, we learn several models that were proposed in the literature to later select the best performing one for the inference of relevance judgments in LOVBench. We choose to experiment with the two best-performing modeling approaches presented in [48], i.e., the User Browsing Model (UBM) [66] and the Dynamic Bayesian Network model (DBN) [44]. As a baseline, we consider a simple Document-based Click-Through Rate model (DCTR) [54]. We randomly select 75% of the logged LOV search sessions for training and hold the remaining 25% out for testing to learn each model[3].

After learning the user click model, we can infer relevance for the query-term pairs contained in the search logs. In general, inferring relevance *Rel* for a query Q and a document (i.e., a term t) from a learned click model is based on user satisfaction probability [48].

$$Rel_{Q,t} = P(S_t = 1 | C_t = 1) * P(C_t = 1 | E_t = 1) \tag{6.1}$$

where S_t, C_t, and E_t are binary random variables corresponding to the user's satisfaction probability, click probability, and examination probability for a term t, respectively. The considered user click models differ as follows. UBM – unlike DBN – does not consider actual user satisfaction in the model, meaning this formula simplifies to the attractiveness probability (i.e., perceived relevance). Another fundamental difference of UBM's and DBN's assumptions concern the examination probability, which in UBM depends on previous clicks and their ranks, while in DBN it depends only on the term position [48]. In DCTR, relevance corresponds to the click-through rate, i.e., how often a term was clicked compared to how often it was shown.

6.3.2.2 Click Prediction Evaluation

As a first step, we evaluate the learned models with common evaluation measures on the held-out test set with regard to click prediction. One evaluation measure for this purpose is the standard log-likelihood [48].

$$LL(M) = \sum_{s \in \mathcal{S}} \sum_{r=1}^{n} \log P_M(C_r = c_r^s | C_{<r} = c_{<r}^s) \tag{6.2}$$

where P_M corresponds to model's M click probability measure at rank r for a session $s \in S$ in the test set, and c_r being the actual click information. We further consider perplexity based on full probability and perplexity gain [48].

$$\begin{aligned} \text{Perplexity}(M) &= \frac{1}{n} \sum_{r=1}^{n} 2^{-\frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} (c_r \log_2 + (1-c_r) \log_2 (1-p_r))} \\ \text{Gain}(M_A, M_B) &= \frac{\text{Perplexity}(M_B) - \text{Perplexity}(M_A)}{\text{Perplexity}(M_B) - 1} \end{aligned} \tag{6.3}$$

where p_r corresponds to the model's predicted click probability ($P_M(C_r = 1|q, t)$).

The results on the model's ability to predict the clicks are summarized in Table 6.2. Both, DBN and UBM outperform the baseline DCTR model. However, UBM clearly shows a better performance for log-likelihood as well as perplexity compared to DBN. We thus find that the learned model using UBM is able to predict clicks well and should be preferred over the others. However, we continue the evaluation based on a comparison with human judgments.

[3] Implementation taken from `https://github.com/markovi/PyClick`

Table 6.2: Evaluation of learned LOV click models with regard to click and relevance prediction. Best performance for each metric is highlighted in bold.

Model	Click			Relevance
	Log-likelihood	Perplexity	Gain	r_{CBRBench}
DCTR	−0.274	1.299	-	0.448
DBN	−0.271	1.227	0.240	0.282
UBM	**−0.173**	**1.187**	**0.375**	**0.613**

6.3.2.3 Relevance Prediction Evaluation

In addition to the previous evaluation on the held-out test set, we leverage existing term relevance judgments from the literature for evaluation purposes. By evaluating the models' performance with regard to their relevance predictions we ensure that inferred relevance labels are accurate and we gain more confidence in the ground truth. In particular, we rely on the expert judgments from CBRBench [38]. CBRBench is composed of ten queries with 34 to 137 term judgments per query. We are able to compare six queries from CBRBench that have overlapping judgments with the relevance predictions of the learned click models based on the results shown in the LOV logs (i.e., *location*, *address*, *organization*, *event*, *music*, and *person*). We use the Pearson Correlation Coefficient r to evaluate the similarity of the total order of terms based on the predicted satisfaction probability with the total order of terms based on the CBRBench judgments.

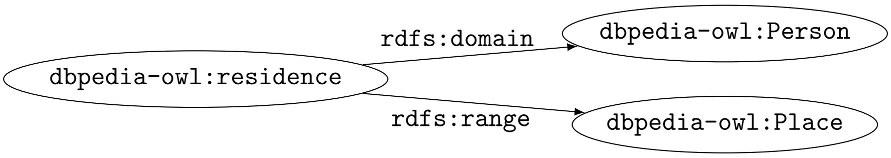
The results of the relevance prediction evaluation are also summarized in Table 6.2, showing the correlation r for each model with respect to the CBRBench judgments. On the one hand, this evaluation step confirms the previous results that the UBM should be preferred over DCTR and DBN. Furthermore, UBM demonstrates a strong correlation with CBRBench's relevance judgments (0.613), meaning that the relevance predictions of the UBM model learned from the LOV logs are close to the expert judgments from CBRBench. We explain the better performance of UBM compared to DCTR and DBN due to its consideration of previous clicks and their position in the same session, based on the experimental findings and intuition presented in [48], where UBM also performs best. Thus, we consider the relevance labels inferred from the UBM model in the LOVBench dataset.

6.4 LOVBench Dataset

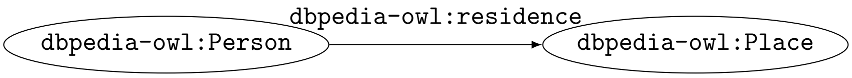
In this section, we introduce the considered ranking features and the sampled ground truth for the proposed ontology ranking benchmark dataset. We present the considered features in LOVBench with a joint conceptualization in Section 6.4.1, introduce our term sampling strategy in Section 6.4.2, and finally summarize the LOVBench dataset in Section 6.4.3.

6.4.1 Feature Description

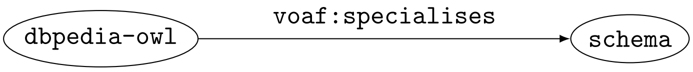
We first introduce the notation used to describe the features in a joint framework. Subsequently, we provide an overview and more details about the considered ranking features.



(a) RDF graph **O**



(b) Ontology graph **G**(**O**)



(c) Repository graph $\mathbf{G}(\mathbf{R}, \{\mathbf{p} \in \text{voaf}\})$

Figure 6.3: Illustration of graph definitions used for ranking.

6.4.1.1 Notation

Let O be an ontology in a repository R. An ontology O is defined by a set of RDF triples `(subject, predicate, object)`, which is referred to as RDF graph. A term t in an ontology can be of type class c[4] or property p[5]. A user query is denoted by Q, with ith word denoted as q_i. The set of ontology terms that matches the query is denoted by $\sigma_{\mathcal{T}}(Q, O)$, where $\mathcal{T}$ corresponds to the type of matched terms, i.e., $\mathcal{T} \in \{t, c, p\}$ (corresponding to all terms, only classes, or only properties, respectively). In addition to the RDF graph O (illustrated in Figure 6.3a), two more graph structures can be derived and used for ranking, namely the ontology graph (Figure 6.3b) and the repository graph (Figure 6.3c). These graph structures allow the application of conventional graph scoring algorithms.

Ontology graph. The ontology graph of O represents classes and properties of the ontology. Unlike the RDF graph, properties are modeled as edges instead of nodes. The ontology graph is denoted by $G(O) = (\mathcal{C}, \mathcal{E}_{c_i,c_j})$, where $\mathcal{C}$ is a set of nodes representing all classes ($c \in O$) and $\mathcal{E}$ is a set of directed edges. These are derived based on the ontologies' properties ($p \in O$) and the semantics of `rdfs:domain` and `rdfs:range`, i.e., $\mathcal{E}_{c_i,c_j} = \{(c_i, c_j) \in O : (p, \texttt{rdfs:domain}, c_i)$ & $(p, \texttt{rdfs:range}, c_j)\}$. A more detailed description of the mapping from O to $G(O)$ can be found in [239].

Repository graph. The repository graph expresses the relationships among ontologies in the repository. It is denoted by $G(R, \mathcal{P}) = (\mathcal{O}, \mathcal{E}_{O_i,O_j})$, where $\mathcal{O}$ is a set of nodes corresponding to the ontologies contained in the repository ($O \in R$) and $\mathcal{E}_{O_i,O_j}$ is a set of edges representing all considered properties $p \in \mathcal{P}$ that exist between ontologies, i.e. $\mathcal{E}_{O_i,O_j} = \{(O_i, O_j) \in R : (O_i, p \in \mathcal{P}, O_j)\}$.

6.4.1.2 Feature Selection

Our strategy for the selection of ranking features for LOVBench is guided by related literature surveys [121, 60, 247], and respectively discovered features are selected based

[4]Considered class types: `rdfs:Class`, `owl:Class`.

[5]Considered property types: `rdf:Property`, `rdfs:Property`, `owl:ObjectProperty`, `owl:DatatypeProperty`, `owl:AnnotationProperty`, `owl:OntologyProperty`.

on the following applicability constraints.

- The feature must rely on a keyword-based query format and rank ontologies or terms. We choose to consider both, term and ontology ranking features in LOVBench because the consideration of the overall quality of an ontology in which a term is defined is often an important factor for reuse [196], and thus, ontology features are often applied for term ranking, such as in CBRBench [38].
- The feature must not depend on metadata information that cannot be derived from the ontology collection itself. Such features are often considered in ontology ranking by formulating scores involving user feedback such as ratings and clicks [141], ontology and term usage in LOD [197], etc. While these features can be very useful for ranking, applying these features to different ontology collections can be problematic because the respective metadata is often not available or difficult to obtain for other ontology collections [120]. Since ranking models evaluated by LOVBench should be applicable to any ontology collection, we choose to exclude them.
- The feature must form a significant distinction to already considered features.
- Priority is given to features for which the complete ranking configuration as proposed in the literature can be replicated.

Other considerations for a fair comparison of ranking models include the constraints that define whether a term matches a query or not. We choose to harmonize the query match for all features independent from the respective feature's original approach. This gives a more accurate query match for all features by considering the meta vocabularies used in the LOV collection. Lastly, some ranking features depend on hyperparameters that need to be set by experts. In LOVBench, we stick to the hyperparameters as suggested in the original source of the feature. However, our large ground truth for the application of LTR further allows us to decompose some of these features (e.g., in case of weighted sums) and implicitly learn the parameters from the data instead.

6.4.1.3 Feature Description

An overview of LOVBench's ranking features, based on the previously introduced notation, is presented in Table 6.3. The subscript after the feature name indicates whether the feature assigns scores to terms (t), ontologies as a whole (O), or only to the query (Q). We would like to note that Q-features cannot be used stand-alone for ranking and that O-features assign the same score for all terms from the same ontology. The feature category indicates the feature's parameters for scoring. We further organize the features in four groups (*query match*, *repository graph analysis*, *ontology graph analysis*, and *RDF graph analysis*), for which we provide more details in the following. As motivated previously, it is possible to use LOVBench for comparisons of rankings using metadata-based features (as an additional group of features), however, it is considered out of scope.

Query match (Feature 1-7). Query match features assess how well the words in the query match the words that describe a term in the ontology. The boolean match (Feature 1), e.g., simply states whether a term is contained in the query match or not and the text relevancy (Feature 4) measures how many words in the query match a term. The original matching features followed in LOV (Feature 2 and 3) are based on a standard BM25 matching score [188]. Feature 2 further assigns weights depending on which property of a term matches the query and Feature 3 is based on properties that describe the ontology [232]. We propose two more features in addition to those used in the state-of-the-art. First, we propose a variation of the class match (Feature 5) for properties (Feature 6). These features are based on different weights for exact and partial matches of query words. Albeit not directly related to matching, we further consider a simple feature that counts the number of words in the query (Feature 7). The following details were considered when extracting query match features for LOVBench.

- A term matches the query (i.e., $t \in \sigma_t$) if at least one word $q_i \in Q$ matches the domain of at least one of the following properties of the term: `rdfs:label`, `dce:title`, `dcterms:title`, `skos:prefLabel`, `rdfs:comment`, `rdfs:description`, `dce:description`, `dcterms:description`, `skos:altLabel`, or the local name of the term's URI. These are the same properties that were considered in the original LOV term property boost [232].
- The exact and partial matches (φ^{exact} and φ^{partial}) of the class and property match (Feature 5 and 6) only consider matches in `rdfs:label`. The hyperparameters are set to $\alpha = 0.6$ and $\beta = 0.4$ [6].

We would like to note that features that use terms or ontologies contained in the query match as input for scoring are not classified as query match features, but assigned to one of the following groups to which the scoring algorithm relates.

Repository graph analysis (Feature 8-10). These features determine the importance of ontologies in the corpus based on the repository graph $G(R, \mathcal{P})$. PageRank has been the most commonly adapted ranking algorithm for this purpose, especially in the domain of ontology search engines [121]. The repository graph is usually constructed based on `owl:imports` statements (Feature 8), however, it has been pointed out that explicit `owl:imports` statements are often missing, and considering implicit imports derived from term URIs appearing in the ontology (Feature 9) showed better performance [37]. We further propose to build a repository graph using ontologies' properties based on the vocabulary of a friend[6] (`voaf`), which in LOV are automatically derived from the ontology collection [232], and to consider this graph as input for PageRank (Feature 10).

- PageRank *pr* for repository graphs $G(R, \mathcal{P})$ is computed as follows [38].

$$pr(O, G(R, \mathcal{P})) = \frac{1-d}{|R|} + \sum_{O_j \in (O_i,\ p \in \mathcal{P},\ O_j)} \frac{pr(O_j, G(R, \mathcal{P}))}{|(O_j,\ p \in \mathcal{P},\ O_i)|} \tag{6.4}$$

  where d corresponds to the damping factor (set to $d = 0.85$). Since PageRank computes very small scores, we multiple these with 10^5.

Ontology graph analysis (Feature 11-16). These features assign scores to terms and ontologies based on the ontology graph $G(O)$. PageRank has also been adapted for this purpose. In particular, [37] proposes a Reversed PageRank approach to identify hubs in an ontology (Feature 11-13). However, this approach is limited to classes since PageRank only assigns scores to nodes and not to edges (i.e., these scores ignore properties). Other considered graph scoring algorithms include *betweenness* (Feature 15), for which we propose a variation for scoring on terms (Feature 14), as well as the semantic similarity measure (Feature 16). The following details are considered for creating LOVBench.

[6] `https://lov.linkeddata.es/vocommons/voaf/v2.3/`

- The betweenness measure scores classes based on the number of shortest paths passing through it [6].

$$betweenness(c, G(O)) = \sum_{c_i \neq c_j \neq c \in O} \frac{\lambda(c_i, c_j(c), G(O))}{\lambda(c_i, c_j, G(O))} \tag{6.5}$$

  where $\lambda(c_i, c_j, G(O))$ corresponds to the total number of shortest paths from c_i to c_j, and $\lambda(c_i, c_j(c), G(O))$ defines the number of shortest paths passing through c.

- Semantic similarity is a measure based on the shortest path between two classes to capture how close these concepts are laid out in the ontology [6].

$$ssm(c_i, c_j, G(O)) = \begin{cases} \frac{1}{|\min(\lambda(c_i, c_j, G(O)))|}, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{6.6}$$

RDF graph analysis (Feature 17-33). Features based on the RDF graph O extract statistics about terms in an ontology, which, e.g., can give an indication about the importance of terms in the repository and to which detail a term is defined. TF–IDF-based measures (Feature 17-25) differ in ontology ranking compared to conventional text retrieval since scoring is adapted to compute the importance of term URIs in the repository [38]. This implies that TF–IDF-like term features (Feature 17-19) are query-independent. TF–IDF ontology features (Feature 20-22), however, are computed based on all terms in an ontology that match the query [38], and thus remain query-dependent. Since the intuition of TF–IDF does not apply to ontology ranking (IDF assigns a low score when a term appears in many ontologies, even though it is a desirable trait), we decompose the measures into separate features (Feature 17-18 and 20-21). In addition to TF–IDF-based features, simple statistics of the RDF graph are also considered for the ranking of terms and ontologies (Feature 26-33). We propose variations of features from the literature which are adapted for ranking of terms instead of ontologies (Feature 26-30) as well as for the consideration of properties instead of classes (Feature 32-33). The following details were considered for LOVBench.

- Term frequency $tf(t, O)$ and inverse document frequency $idf(t, R)$ are computed as follows [38].

$$\begin{aligned} tf(t, O) &= 0.5 + \frac{0.5 * f(t, O)}{\max(\{f(t_i, O) : t_i \in O\})} \\ idf(t, R) &= \log\left(\frac{|R|}{|\{O : t \in O, O \in R\}|}\right) \end{aligned} \tag{6.7}$$

  where $f(t, O)$ is the frequency of a term t in ontology O.

- The adapted notion of BM25 for ranking of terms t is defined as follows [38].

$$bm25(t, O, R) = \mathrm{IDF}_t(t, R) * \frac{\mathrm{TF}_t(t, O) * k + 1}{\mathrm{TF}_t(t, O) + k * \left(1 - b + b * \frac{|O|}{avgos(R)}\right)} \tag{6.8}$$

  where k and b are hyperparameters ($k = 2.0$, $b = 0.75$), the ontology size $|O|$ is computed by its number of triples, and $avgos(R)$ is the average ontology size in the repository.

Table 6.3: LOVBench Ranking Features

ID	Feature	Cat.	Description	Ref.
Query match				
1	Boolean match$_t$	Q, t	$1, \text{if } t \in \sigma_t; 0, \text{otherwise}$	[38]
2	Match-boost$_t$	Q, t, R	$bm25(Q, t, R) + boost(\sigma_t)$	[232]
3	Match-descr.$_O$	Q, O, R	$bm25(Q, O, R)$	[232]
4	Text relevancy$_t$	Q, t	$\|\{q_i \in Q : \sigma_t \neq \varnothing\}\|$	[37]
5	Class match$_O$	Q, O	$\sum\limits_{q_i \in Q} \alpha * \|\varphi_c^{\text{exact}}\| + \beta * \|\varphi_c^{\text{partial}}\|$	[6]
6	Prop. match$_O$	Q, O	$\sum\limits_{q_i \in Q} \alpha * \|\varphi_p^{\text{exact}}\| + \beta * \|\varphi_p^{\text{partial}}\|$	
7	Query length$_Q$	Q	$\|\{q_i \in Q\}\|$	
Repository graph analysis				
8	PR-imports$_O$	O, R	$pr(O, G(R, \{\texttt{owl:imports}\}))$	[38]
9	PR-implicit$_O$	O, R	$pr(O, G(R, \{\textit{implicit imports}\}))$	[38]
10	PR-voaf$_O$	O, R	$pr(O, G(R, \{p \in \texttt{voaf}\}))$	
Ontology graph analysis				
11	Hub$_t$	c, O	$pr(c, G(O_{\text{reversed}}))$	[37]
12	Max hub$_O$	O	$\max(\{\text{Hub}_t : c \in O\}\})$	[37]
13	Min hub$_O$	O	$\min(\{\text{Hub}_t : c \in O\}\})$	[37]
14	Betweenness$_t$	t	$betweenness(t, G(O))$	[6]*
15	Betweenness$_O$	Q, O	$\frac{1}{\|\sigma_c\|} \sum_{c \in \sigma_c} \text{Betweenness}_t$	[6]
16	Semantic sim.$_O$	Q, O	$\frac{1}{\|(c_i, c_j)\|} \sum\limits_{c_i, c_j \in \sigma_c} ssm(c_i, c_j, G(O))$	[6]
RDF graph analysis				
17	TF$_t$	t, O	$tf(t, O)$	[38]*,**
18	IDF$_t$	t, R	$idf(t, R)$	[38]*,**
19	TF-IDF$_t$	t, O	$\text{TF}_t * \text{IDF}_t$	[38]*
20	TF$_O$	Q, O	$\sum_{t \in \sigma_t} \text{TF}_t$	[38]**
21	IDF$_O$	Q, R	$\sum_{t \in \sigma_t} \text{IDF}_t$	[38]**
22	TF-IDF$_O$	Q, O, R	$\sum_{t \in \sigma_t} \text{TF-IDF}_t$	[38]
23	BM25$_t$	t, O, R	$bm25(t, O, R)$	[38]*
24	BM25$_O$	Q, O, R	$\sum_{t \in \sigma_t} \text{BM25}_t$	[38]
25	VSM$_O$	Q, O, R	$vsm(Q, O, R)$	[38]
26	Subclasses$_t$	c	$\|subclasses(c)\|$	[6]*,**
27	Superclasses$_t$	c	$\|superclasses(c)\|$	[6]*,**
28	Relations$_t$	c	$\|relations(c)\|$	[6]*,**
29	Siblings$_t$	c	$\|siblings(c)\|$	[6]*,**
30	Density$_t$	c	$w_1 * \text{Subclasses}_t + w_2 * \text{Superclasses}_t + w_3 * \text{Relations}_t + w_4 * \text{Siblings}_t$	[6]*
31	Density$_O$	Q, O	$\frac{1}{\|\sigma_c\|} \sum_{c \in \sigma_c} \text{Density}_t$	[6]
32	Subproperties$_t$	p	$\|subproperties(p)\|$	
33	Superprop.$_t$	p	$\|superproperties(p)\|$	

* Adapted for terms, originally only proposed for ontology ranking.
** Originally not considered as individual feature.

- The Vector Space Model (VSM) [195] feature (Feature 25), adapted to ontology ranking, measures similarity of query and ontology using *tf* and *idf* to compute the weights of query words q_i in the query and ontology [38].

$$vsm(Q,O,R) = \frac{\sum_{q_i \in Q} \left(\sum_{t \in \sigma_t(q_i,O)} (\text{TF-IDF}_t(q_i,t)) * \textit{tf-idf}_Q(q_i,Q,R) \right)}{\sqrt{\sum_{t_i \in O} (\text{TF-IDF}_t(t_i,O))^2} * \sqrt{\sum_{q_i \in Q} (\textit{tf-idf}_Q(q_i,Q,R))^2}} \tag{6.9}$$

where *tf-idf*$_Q$ of a query word q_i is defined as follows.

$$\begin{aligned} \textit{tf-idf}_Q(q_i,Q,R) = & \frac{f(q_i,Q)}{\max(\{f(q_j,Q) : q_j \in Q\})} * \\ & \log\left(\frac{|R|}{|\{O : t \in O, t \in \sigma_t(Q,O)\}|}\right) \end{aligned} \tag{6.10}$$

- The density (Feature 30) depends on a weighted sum, for which the original weights were used: $w_1 = 1.0$, $w_2 = 0.25$, $w_3 = 0.5$, and $w_4 = 0.5$ [6].

In summary, we presented the selected state-of-the-art ranking features and proposed variations and features that are motivated by our observations made from the LOV search logs. In the next step, we describe our strategy to select samples for the ground truth.

6.4.2 Term Sampling

When building a ground truth, it is usually impractical to judge and extract features for all terms in the repository [132]. In the context of LTR, the chosen strategy of sampling some documents for each query for learning will impact the observed effectiveness of the learned ranking model, referred to as *sample selection bias* [134, 150]. Given the discussions of document sampling's impact [134], our goal is to ensure that the LOVBench dataset contains a similar distribution of relevance labels and average sample size compared to well-known state-of-the-art datasets for conventional ad-hoc Web retrieval tasks. We follow a similar strategy to the one presented in LETOR [179], by (i) considering all query-term pairs without judgment as non-relevant and (ii) selecting samples from an existing ranking, i.e., the term search results of LOV. For each query of the LOV search logs, the relevance of the first ten terms is inferred from the best-performing click model using UBM (see Section 6.3.2), and an additional amount of non-relevant terms is randomly sampled from the remaining query match.

As a popular convention, we map the relevance inferred from the click model, which is measured in terms of satisfaction probability ($0 \leq Rel_{Q,t} \leq 1$), to relevance labels $l_{Q,t}$ on a scale from 0-4, and further consider a junk grade (-2). We denote the set of labeled data for query Q and term t by $\mathcal{L} = \{(Q,t,l_{Q,t})\}$, the standard input for supervised LTR settings, and refer to $\mathcal{L}$ with all relevance predictions as *ground truth.* As shown in Table 6.4, this approach results in a dataset with a similar distribution of relevance judgments compared to established LTR datasets, such as TREC Web 2013 & 2014 [51, 52] and a selection of ClueWeb12[7] entries as presented in [4]. On the other hand, it reveals a different distribution for CBRBench, which is a potential explanation for the differences in our inferred relevance judgments and those made by

[7] `https://lemurproject.org/clueweb12/`

Table 6.4: Breakdown of relevance labels in LOVBench, well-known ad-hoc Web retrieval datasets, and CBRBench.

Label	LOVBench	TREC Web	ClueWeb12	CBRBench
Total	73,950 + 110,274	28,906	5,392	819
Very high (4)	0.34%	0.14%	0.07%	4.64%
High (3)	1.63%	1.42%	1.15%	10.01%
Low (2)	7.26%	8.77%	5.47%	19.90%
Very Low (1)	24.10%	23.64%	20.83%	25.76%
Non (0)	66.59%	63.31%	68.64%	39.68%
Junk (−2)	0.06%	2.73%	3.84%	0.00%

experts from CBRBench. The authors of CBRBench only consider a term as relevant with a score of 2 or higher [38], while we follow the common practice to consider a score of 1 or higher as relevant. Finally, LOVBench is composed of 73,950 judgments inferred through the user click model and 110,274 randomly sampled non-relevant judgments, with an average sample size of ~ 26 judgments per query.

6.4.3 Final LOVBench Dataset

In summary, the LOVBench dataset contains: (i) extracted ranking features (Section 6.4.1.3) and (ii) inferred relevance judgments (Section 6.4.2) for 184,224 query-term pairs. We provide a single csv-file containing the query-term pairs and their relevance judgments which can be used as ground truth for evaluation purposes. Moreover, the full dataset (including extracted features) is provided in a format for LTR experiments. It is randomly split by query into five equal-sized partitions, which in turn are used to derive five folds with three partitions for training and the remaining two for validation and testing.

6.5 Empirical Evaluation

In this section, we apply LOVBench to evaluate several ranking configurations using LTR. We first describe our experiment setup and then present our experimental results.

6.5.1 Experimental Setup

The experiments are based on the standard framework for LTR evaluation [132]. In the following, we describe the considered feature configurations, baselines and evaluation metrics. We rely on the previously introduced LOVBench dataset for the evaluation.

Configurations. We experiment with feature configurations that are associated with three different groups: baseline, literature, and variations. An overview of all configurations subject to the experiments is given in Table 6.5. The purpose of our baseline (a standard Lucene search on terms' `rdfs:label` properties) is to provide the performance of a straight-forward search approach as a reference point. We then compare several ranking models as they have been proposed in the literature with the baseline using the feature sets from AKTiveRank [6], DWRank [37], and CBRBench [38]. Lastly, we propose variations that are designed to meet the requirements observed in the LOV search logs. We compare three models to the baseline and literature:

Table 6.5: Overview of feature configurations. The feature IDs correspond to those presented in Table 6.3.

Category	Name	Feature configuration	Count
Baseline	Baseline	Lucene search (`rdfs:label`)	1
Literature	DWRank [37]	4, 9, 11-13	5
	AKTiveRank [6]	5, 15-16, 31	4
	CBRBench [38]	1, 5, 9, 15-16, 22, 24-25, 31	9
Variations	LOV-based	2-3	2
	LOVBench full	1-33	33
	LOVBench light	2, 10, 14, 17-19, 26-29, 32-33	12

first, a simple model only using current LOV features as a reference point (LOV-based), second, a model using all features of LOVBench (LOVBench full), and third, a model with features that are chosen based on low computational complexity, high informativeness and meeting the LOV requirements (LOVBench light). In total, we evaluate ranking performance of seven different configurations with up to 33 different features.

Learning to rank. Each model configuration is trained with well-known LTR techniques. We experiment with both, a pairwise approach (RankNet [32]) as well as a listwise approach (AdaRank [243])[8]. Evaluation and comparison to the baseline is based on five-fold cross-validation.

Metrics. We rely on standard evaluation metrics for information retrieval. We measure the retrieval performance in terms of multi-valued relevance based on NDCG [111]. Motivated by the discussion in Section 6.3.1, we report the NDCG for the first 3, 5, and 10 ranked terms. We further report MAP [13] as a consideration of ranking models' effectiveness, which, however, only considers binary relevance.

6.5.2 Experimental Results

In this section, we present the experimental results and compare each configuration to the baseline.

6.5.2.1 Originals

In Table 6.6, we compare the performance of several feature configurations using two LTR algorithms with the baseline. The first row shows the baseline performance based on a Lucene search. The first group of configurations (DWRank, AKTiveRank, CBRBench) for both LTR algorithms shows the performance of original models from the literature.

When comparing the results of the original configuration with the baseline, we observe that all configurations manage to significantly improve the baseline performance in all metrics. Compared with each other, all models show similar performance, albeit DWRank performs slightly better when using RankNet and AKTiveRank performs slightly better when using AdaRank. Another interesting observation is that the additional features considered in CBRBench, which also includes the features proposed in AKTiveRank, barely improve (RankNet) or even harm (AdaRank) the ranking perfor-

[8] Implementation taken from `https://sourceforge.net/p/lemur/wiki/RankLib`

Table 6.6: Results for ranking models with considered feature configurations using LOVBench. Best results for each algorithm and metric are highlighted in bold. All configurations significantly improve the performance in all metrics compared to the baseline (p-value $\leq \mathbf{0.05}$).

LTR	Configuration	NDCG@3	NDCG@5	NDCG@10	MAP
-	Baseline	0.261	0.299	0.358	0.433
RankNet	DWRank	0.497	0.508	0.536	0.572
	AKTiveRank	0.459	0.469	0.499	0.563
	CBRBench	0.480	0.484	0.512	0.564
	LOV-based	0.646	0.688	0.724	0.717
	LOVBench full	**0.841**	**0.868**	**0.883**	**0.845**
	LOVBench light	0.762	0.781	0.803	0.806
AdaRank	DWRank	0.372	0.382	0.404	0.481
	AKTiveRank	0.454	0.454	0.478	0.548
	CBRBench	0.378	0.379	0.411	0.492
	LOV-based	0.638	0.679	0.716	0.710
	LOVBench full	**0.881**	**0.902**	**0.911**	**0.918**
	LOVBench light	0.780	0.825	0.860	0.871

mance. This is surprising, given that the comparison in CBRBench [38] based on single features shows some of these features to perform well. This indicates that ranking models should always be evaluated with all features and weights taken into account, since individually poorly performing features can still perform well when combined using LTR.

6.5.2.2 Variations

The second group for each algorithm (LOV-based, LOVBench full, LOVBench light) in Table 6.6 shows the results of the adapted models.

First, we can observe that all three configurations that use the original LOV term matching feature (Feature 2) significantly improve the performance of the baseline as well as the models from the literature. However, we need to keep in mind that the sampled ground truth is biased on the existing ranking of LOV, which includes Feature 2 [232]. Moreover, when comparing the LOVBench full model with the simple LOV-based configuration, we observe that the additional features in LOVBench significantly improve the performance. Given that real-time computational requirements can be a major concern, we are also interested in increasing the ranking performance with a reduced number of features compared to the full configuration. As the results of the LOVBench light configuration show, the smaller feature set is also able to significantly improve ranking performance compared to the LOV-based model, albeit not as much as the full model. All these observations hold true for both learning algorithms, RankNet and AdaRank.

6.6 Discussion

In this section, we discuss the practical implications of the results and state the limitations of our approach.

6.6.1 Practical Recommendations

One of our key findings is that ranking features should be designed with regard to the actual user behavior of the targeted domain. The performance results of the considered literature models can be explained with the mismatch of the assumptions for these ranking models and the user behavior we observe in LOV (cf. Section 6.3.1). First, the original AKTiveRank configuration does not include any term features, meaning that if the ranking contains multiple terms from the same ontology, these receive the same score. Second, in DWRank, three out of five features (Feature 11-13) relate to the hub score, which only scores classes, meaning that this configuration lacks the ability to rank properties, which we have found in LOV to be equally important as the ranking of classes. Furthermore, the text relevancy feature that is also part of this configuration (Feature 4) assumes multiple words in the query, which in LOV is often not the case. Third, CBRBench is, similar to AKTiveRank, mostly composed of ontology features, and further of many complex features with predefined hyperparameters, not allowing the LTR algorithms to learn these from the data. As shown by our experiments, adapting the ranking features to these requirements significantly improved the performance. Our results also demonstrate that increasing the number of ranking features, given a large enough dataset like LOVBench, allows increasing ranking performance. However, in practice it is often necessary to form a reasonable trade-off between effectiveness and computation cost. This especially concerns query-dependent features that cannot be pre-computed and have to be extracted at run-time.

As previously mentioned, the experimental results need to be interpreted in the context of the biases imposed by the current LOV ranking from which the relevance labels for the ground truth are derived. The LOV search relies on a Lucene-based match with property boost and a popularity score measured in term of usages in LOD datasets [232]. Thus, it is reasonable that relevance features and qualitative importance features such as PageRank perform better on our dataset than graph-structural criteria such as density.

In conclusion, the following considerations should be taken into account when designing ontology ranking models, which were also followed for the LOVBench light configuration:

1. mixing and prioritizing of ranking granularities depending on the platform (ontologies/terms and classes/properties),
2. diverse coverage of all feature categories (query match, repository graph, ontology graph and RDF graph analysis),
3. preferring decomposed features when possible,
4. being aware of the user behavior of targeted platform,
5. the number of query match features should be minimized.

Lastly, LOVBench can be used not only to evaluate, but also to train ranking models and integrate these for re-ranking of ontology terms in other applications with different ontology collections. In this case, we would like to highlight that the query match constraint should be adjusted to the meta-vocabularies considered in the respective ontology collection and all features need to be extractable for the LOV ontology collection.

6.6.2 Limitations

The applicability of LOVBench for the evaluation of ontology rankings is limited as follows. First, only ranking models that use a keyword-based query format to rank ontology terms can be evaluated. Albeit this is the most popular approach in existing search interfaces [121], this means that LOVBench cannot be considered as a general benchmark in the broader context of ontology reuse. E.g., in some tools other search interfaces and rank granularities are used (such as the ranking of the best combination of ontologies [142]). However, the current lack of search logs from respective platforms hinders the application of our approach to other search interfaces. We believe that the adoption of our approach to other ontology search platforms could ultimately result in a landscape of benchmarks for all variations. Second, applying LOVBench for the evaluation of ranking models that include features that rely on metadata information such as user ratings can be difficult, as this metadata needs to be collected for the LOV collection. However, using metadata-based features in the ranking model in general can be problematic, depending on the way the metadata is collected. Instead, sufficient and well-designed ontology and term features directly derived from the ontology collection might be able to substitute such features [120].

6.7 Conclusion

In this chapter, we address the problem of ontology ranking evaluations and comparisons in LTR settings. We analyze logged user interactions of a real-world ontology search platform (LOV) and use this implicit user feedback to infer relevance judgments for query-term pairs. Our evaluation shows that inferred relevance judgments are close to those made by human experts. We create the LOVBench dataset that comprises 184,224 relevance judgments for 7,395 queries and considers 33 different ontology ranking features. We then evaluate and compare three state-of-the-art ranking models from the literature based on this ground truth. We explain the results in the context of the observed user behavior in LOV and propose variations and configurations that outperform the baseline. The LOVBench dataset, the code for the extraction of selected features, the code to run the experiments presented in this chapter, and the cleaned search logs collected from LOV are available online[9].

This chapter concludes the part on efficient ontology recommendation. The following chapter presents an IoT ecosystem application that demonstrates the integration of ontology recommendation in the case of IoT data stream publication.

[9] LOVBench resources: `https://github.com/nut-hatch/LOVBench`

Part III

IoT Ecosystem Application and Conclusion

Chapter 7

IoT Ecosystem Application: Semantic Data Stream Publication

A consensus on ontologies is a necessary condition for semantic interoperability in IoT ecosystems, however, this requires efficient tool support for data annotation and integration within such ecosystems to enable efficient processes and foster wide adoption of the approach. Often, the complexity of Semantic Web approaches and ontologies may form an additional burden. In this chapter, we propose a productivity tool that supports users in publishing semantically annotated IoT data streams. The approach incorporates ontology recommendation in its design, and expands the discussion in this dissertation to other interoperability levels. The practicability of the tool and the resulting IoT gateway is evaluated based on a smart parking scenario within the bIoTope IoT ecosystem. This chapter is based on the work that has been presented in the following paper:

- Niklas Kolbe, Jérémy Robert, Sylvain Kubler, and Yves Le Traon. "PROFICIENT: Productivity Tool for Semantic Interoperability in an Open IoT Ecosystem". In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*. ACM. 2017. DOI: `10.1145/3144457.3144479`

Contents

7.1 Introduction

Recent research initiatives started to investigate open IoT ecosystems that offer provisions to efficiently consume data and other digital services, i.e., to access, discover, aggregate, and semantically understand – both from a human and machine perspective – heterogeneous information sources from various platforms [129]. However, IoT platforms and systems remain isolated silos [234] as there is no established standardized open API that is widely accepted and used by the IoT community. Today's Web consists of a huge amount of custom APIs, and similarly, mobile and IoT services join the so-called *Web API Economy*, thus facing issues related to Web service design and are expected to contribute to the growth of the number of available open Web APIs [105, 221].

This chapter proposes a productivity tool called PROFICIENT to support IoT data/service providers in wrapping custom interfaces with an open and standardized API incorporating ontology recommendation. We rely on the respective technologies for IoT data and service provisioning proposed by an IoT ecosystem project named bIoTope[1] [129], which are introduced later in this chapter. This enables actors to join (if desired) the open IoT ecosystem and gain benefits such as better visibility, new collaboration opportunities and revenues. A prototype of PROFICIENT is developed and presented in this chapter, which is discussed in the context of the LCIM stack (cf. Chapter 1). Ontology recommendations are integrated from the LOV platform since no sufficient ORT dedicated to IoT ontologies exists. Furthermore, the approach is illustrated with a smart parking use case, and the generated IoT gateway is evaluated through a performance analysis.

This chapter is structured as follows. Section 7.2 introduces the conceptual architecture of the proposed productivity tool, whose practicability is demonstrated in Section 7.3 by applying it to a smart parking use case. The proposed approach is finally discussed in Section 7.4; the conclusion follows.

7.2 Productivity Tool Approach

Composing and maintaining IoT services that consume IoT data coming from heterogeneous and custom systems/interfaces is a very complex task. The combination of different protocols (e.g., HTTP, MQTT, XMPP), serializations (e.g., JSON, CSV, XML), and semantic models (e.g., UML models, standard specifications, RDF-based ontologies) impose huge efforts on the consumer to understand, parse, transform and aggregate information for processing. Formally, the complexity can be denoted by

$$c = \frac{n(n-1)}{2} \tag{7.1}$$

where n represents the numbers of different accessed APIs. Therefore, the effort for maintaining the integration of APIs grows exponentially. The same issue has been identified for protocol translation [63] and was presented as the *industrial IoT connectivity challenge* [184].

PROFICIENT is intended to overcome parts of this problem, whose primary objective is to provide data/service providers with a semi-automated solution to easily develop a standardized façade on top of their proprietary interfaces. This is a prerequisite to join and benefit from IoT ecosystem features (e.g., enhanced data/service discovery

[1] `https://biotope-project.eu/`

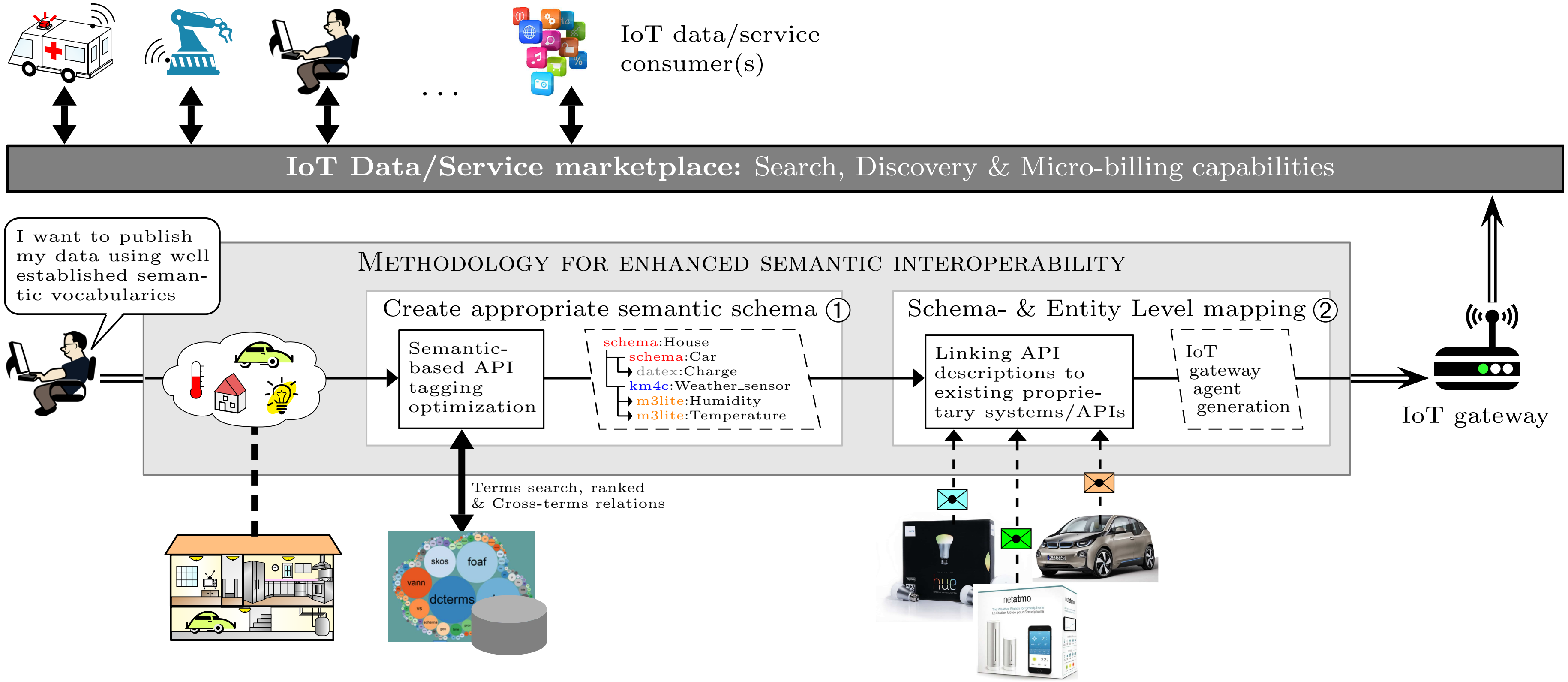


Figure 7.1: Overview of PROFICIENT and associated components.

capabilities, micro-payment opportunities, etc.), like the ones developed through the IoT-EPI initiative [129]. The key motivation is to reduce the development effort (i.e., costs) to create a standardized IoT gateway, and incentivize data and service providers to join open innovation marketplaces. The proposed tool aims to hide the technical complexity of achieving semantic M2M interoperability from the user, which is key to improve user acceptance of semantic-based approaches to a broader audience [18, 202]. The next section provides greater insight into the underlying building blocks of PROFICIENT.

The API harmonization process of the productivity tool is depicted in Figure 7.1, which is a two-step approach denoted by ① and ② in the figure. The first step consists of creating a semantic data structure, while the subsequent step consists in creating a schema- and entity-level mapping of the proprietary data to the newly created data structure. These two steps are further discussed hereinafter:

Step 1: Defining a semantic-based data structure. As a first step, the provider is expected to describe the data/service that is intended for publication. To simplify the usage for users with no experience in Semantic Web technologies, the structure is represented in a tree format. This approach is inspired by the presentation of more popular ontologies like *schema.org* and the *JSON-LD* representation. Other standards and data models could be used in a similar manner to create the semantic schema. The tool supports the selection of ontology terms based on string searches by accessing repositories of semantic ontologies. Exploration of attached elements of a chosen ontology term should also be abstracted from the underlying concept, and selected terms may be added to any part of the targeted schema tree. The example in Figure 7.1 shows a user who intends to publish information about a smart home and creates a tree structure of semantic terms related to his/her facilities/Things (e.g., House, Car, etc. in ①).

Step 2: Defining a schema- and entity-level mapping. In this second step, the assumption is made that the user is already able to access the data (e.g., in the local network or through already implemented Web gateways). To put it another way, the tool's user has to specify the access to the data sources he/she would like to expose to the WoT. Given this assumption, the end-user needs to perform a mapping between the existing data sources and elements of the semantically annotated tree resulting from step 1. These mappings can be of the ones described in Table 7.1. In the second part of the mapping (cf. ② in Figure 7.1), specific entity-based configurations can be made. This could include the specification of transformation rules for certain objects, such as exemption of certain data objects from publication and addition of metadata.

The goal of creating the semantic schema and specifying mappings is to generate a deployable image of an IoT gateway agent which is in charge of pulling the data from the proprietary APIs (e.g., the Netatmo weather station or car examples in Figure 7.1) and to perform the transformations/aggregation of this data, and ultimately publishing the harmonized data to the WoT. From an IoT ecosystem viewpoint, this gateway and the exposed data/services could then be automatically indexed by IoT search engines, be available for trade through IoT service marketplaces (e.g., as the one presented in [129]), etc. All this is illustrated at the top of Figure 7.1.

Table 7.1: Types of Schema- and Entity-level Mappings

Type of mapping	I/O	Description
Simple mapping	$(1:1)$	One term of the targeted schema is mapped to only one property from the proprietary format. Transformation rules, e.g., include conditional expressions to transform proprietary values to ontology terms.
Splitting	$(1:n)$	When mapping a property from the proprietary format to multiple terms of the target schema, a splitting occurs and transformation rules for all terms of the target schema need to be defined. This case could, e.g., occur if coordinates are represented as two comma-separated values in one string, but the targeted semantic schema requires it to be split explicitly to longitude and latitude. Easy-to-use splitting rules can be defined via techniques such as tokenizing the string based on delimiters, based on regular expressions, etc.
Aggregation	$(m:1)$	Aggregation forms the counterpart to splitting, i.e., it occurs when multiple properties are linked to a single term of the target schema. The transformation rule in this case needs to define how to combine the values from different properties (e.g., concatenating two or more values, applying mathematical operations).

7.3 Smart Parking Use Case

The context of the proposed use case falls within the scope of the bIoTope H2020 project, which is part of the IoT-EPI initiative [129]. The bIoTope ecosystem is built upon three building blocks that aim to form a trade-off between RESTful principles using open standards (while also supporting remote procedure calls for heavier Web services) and developing ecosystem components that provide interoperability among all levels under IoT requirements. These three building blocks are briefly introduced in the following.

- Open Data Format (O-DF)[2] standard: It defines a hierarchical data structure of objects which are comprised of InfoItems with values and potentially associated metadata. In addition to O-DF, that solely defines the taxonomy of the data, data models and ontologies are used to define the meaning of the objects and InfoItems.
- Open Messaging Interface (O-MI)[3] standard: It acts as a messaging interface that defines how to call the services, either with resource-oriented requests like read, write and subscribe, or by remote procedure calls. O-MI, in combination with O-DF, forms a service description and is thus applied to achieve pragmatic interoperability.
- IoT service marketplace [129]: It holds a repository of available O-MI/O-DF services and their specifications. Based on the integration of ontologies, it is possible to discover relevant data and services, which can then be accessed in a peer-to-peer fashion in a common publish-find-bind manner. Consumers are also able to track changes in the state of published services, which allows for dynamic interoperability through the marketplace.

[2] `https://www2.opengroup.org/ogsys/catalog/C14A`
[3] `https://www2.opengroup.org/ogsys/catalog/C14B`

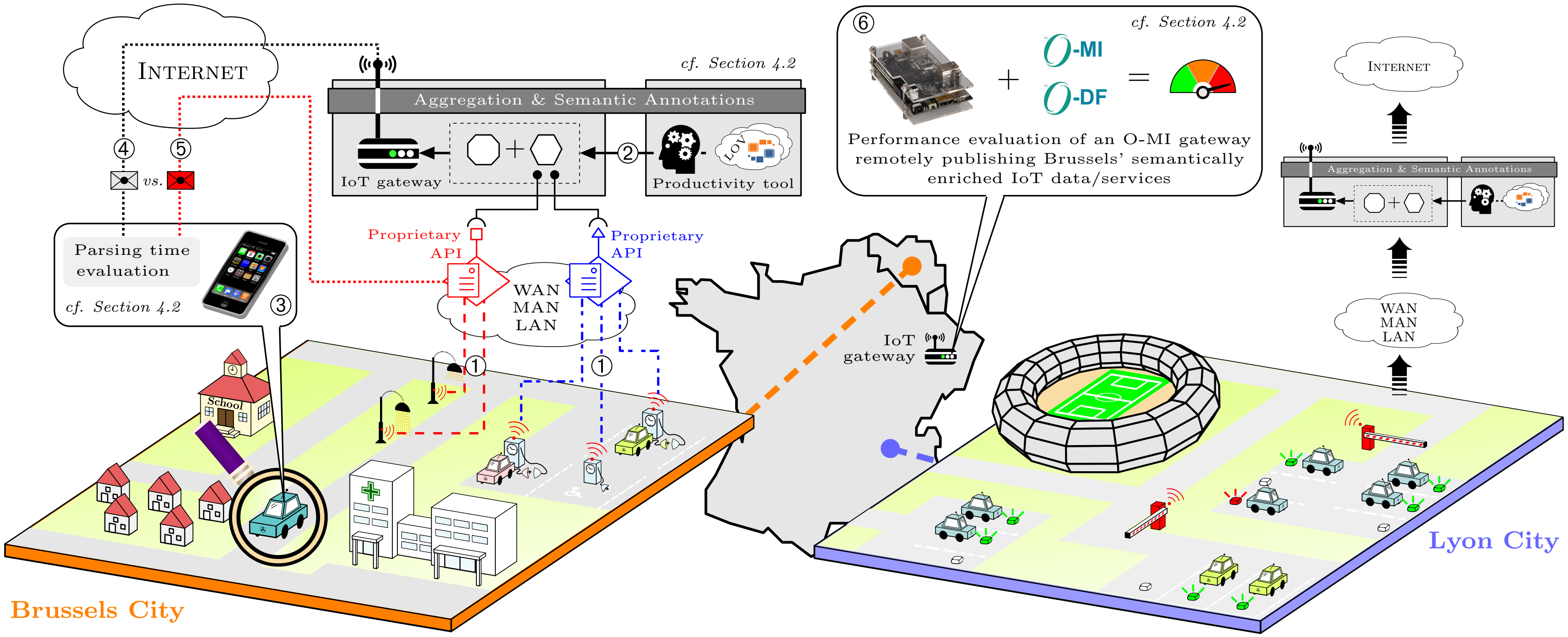


Figure 7.2: Smart parking use case benefiting from productivity tool and the bIoTope building blocks.

In order to demonstrate the practicability of our productivity tool, a smart parking use case has been defined and implemented. This use case extends the one presented in [122], and is illustrated in Figure 7.2. Considered is a scenario in two distinct smart cities, namely Grand Lyon and Brussels Region, both being official partners of the bIoTope project. Step ① in Figure 7.2 illustrates how various data providers of smart things (e.g., of parking sensors or charging stations) in the city expose the data through traditional, proprietary APIs. In step ②, PROFICIENT is used to create a standardized gateway around these APIs. In this demonstrator, two O-MI gateway agents – exposing information of parking facilities in Lyon and in Brussels – are deployed based on the productivity tool prototype (which is presented in following Section 7.3.1). The two existing formats of the parking data (custom JSON-based model and the Datex II[4] vocabulary defined in XML) are mapped to the MobiVoc ontology. Step ③ shows the implementation of an IoT service that relies on the published data, namely a service that is able to discover available parking data and gives recommendations to drivers for best parking locations based on the location and other vehicle-related features. Step ④ shows the bIoTope service flow through the O-MI gateway, whereas step ⑤ shows the traditional way of collecting data from vendor lock-in systems and proprietary APIs.

In the following, the implementation of PROFICIENT is presented in Section 7.3.1. A performance evaluation of the generated O-MI gateway is carried out in Section 7.3.2. This performance evaluation is briefly illustrated in Figure 7.2 as well, for which an O-MI gateway has been hosted in Metz, France (step ⑥). Furthermore, the two different approaches to access IoT data/services (i.e., ④ vs. ⑤) are assessed from a client perspective by comparing the parsing time of the message payloads (cf. ③ in Figure 7.2).

7.3.1 Prototype

The productivity tool concepts presented earlier in Section 7.2 are implemented as a prototype to meet the bIoTope requirements. This implies that O-DF is the targeted format and that the generated IoT gateway agent pushes the O-DF structured data into an O-MI server node. However, the internal representation is based on a generic semantic format, which is JSON-LD.

The implementation of the PROFICIENT prototype is depicted in Figure 7.3. Steps 1 and 2 are illustrated through two distinct screenshots of the Web interface of the productivity tool. In step 1, the user can define the targeted semantic schema(s) of the data to be published by accessing ontology terms from the LOV repository (cf. Section 3.2). The user is able to add terms individually or browse through attached properties and add parts of the tree structure to the targeted schema. An example for a schema of parking data is given for step 1. Subsequently, through step 2, the user is able to link the proprietary API/schema to the targeted schema. Different sources can be defined for the mapping; the example shows a proprietary JSON file that contains information about parking facilities in Lyon. The tool is able to automatically suggest mappings based on a similarity measure of the source string and the ontology terms.

The third screenshot in Figure 7.3 shows the Web interface of a running instance of the O-MI reference implementation[5]. The data is pushed to the node by a generated agent, whose behavior is determined through the defined schema, data sources, mappings, and configurations in PROFICIENT. The final export is a Docker image

[4] `http://www.datex2.eu/`

[5] `https://github.com/AaltoAsia/O-MI`

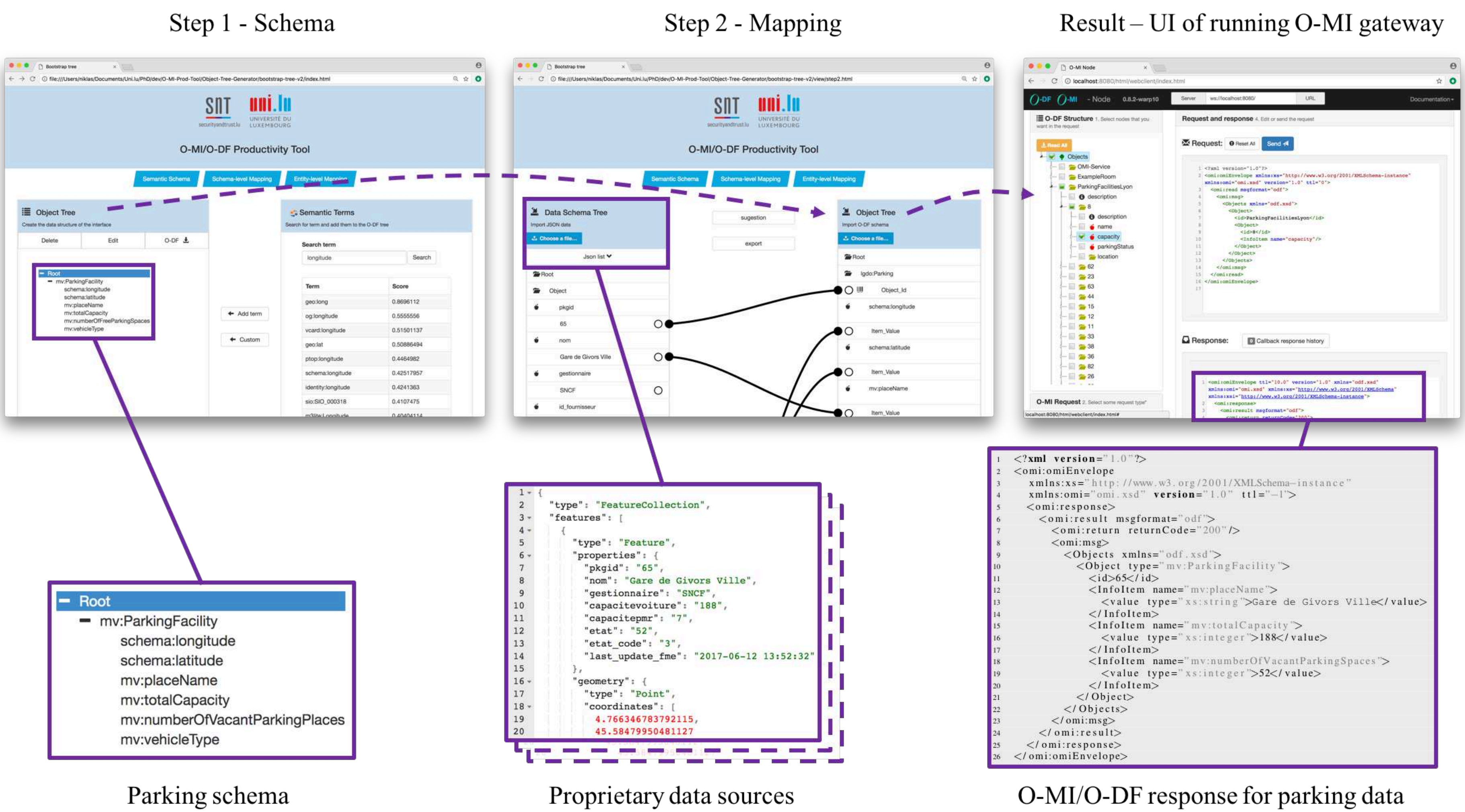


Figure 7.3: PROFICIENT implementation, from proprietary data sources to enriched O-DF published by the generated O-MI agent.

including the setup for the O-MI agent and the O-MI node reference implementation, which is thus ready for immediate deployment to be hosted as an IoT gateway.

O-DF is a tree-based data model and thus not designed to represent RDF-based annotations. However, semantic tags can be added in the O-DF payload by using the *type* attribute of *Objects* and *InfoItems*. These semantic tags are used for the discovery of published data at the IoT service marketplace. An example of the resulting O-DF structure, which is published through the generated O-MI agent of PROFICIENT, is shown in the bottom right corner of Figure 7.3. It shows the O-MI/O-DF response of a *read* request of some parking facility properties.

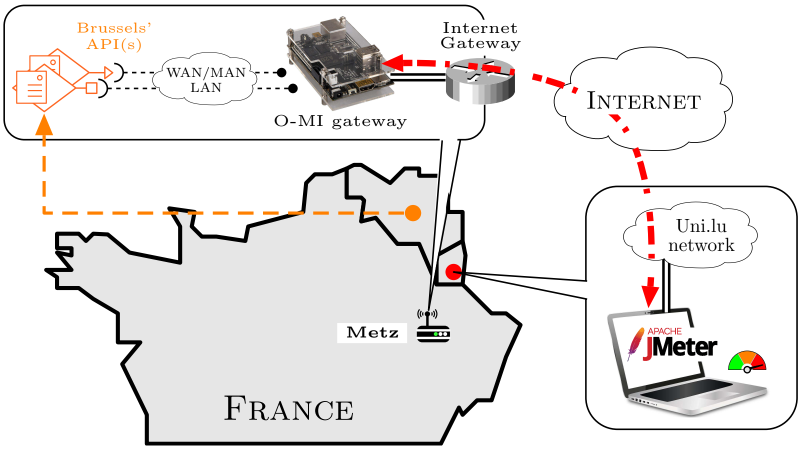


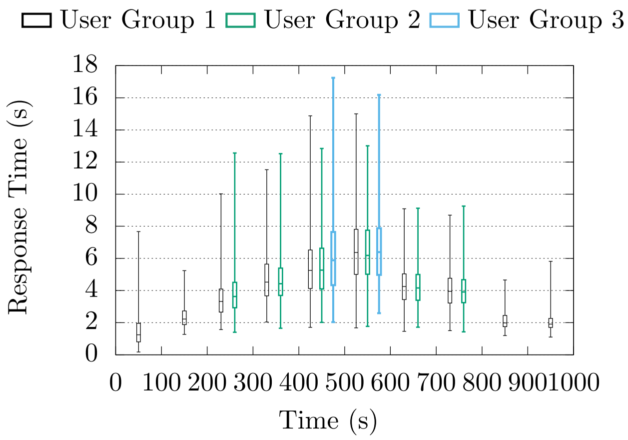
Figure 7.4: Experimental setup for performance assessments.

7.3.2 Performance Evaluation

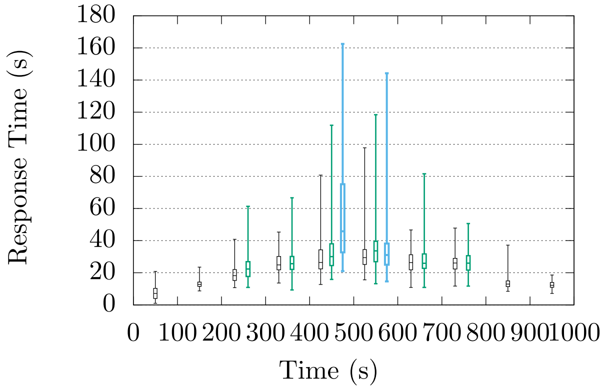
The previously presented smart parking use case is considered to assess the performance and scalability of the O-MI gateway. More concretely, the objective is to evaluate the feasibility to deploy such gateways on resource-constrained devices. The experimental setup is depicted in Figure 7.4. The O-MI server (version 0.8.2 of the reference implementation) is set-up on a resource-constrained device, a *cubieboard*, with the following features: i) *CPU:* 1 ARMv7 Processor rev 2 @[624 – 1008] MHz; ii) *operating system:* ARMBIAN 5.25 stable Debian GNU/ Linux 8 (jessie); iii) *memory:* 1GB. It is hosted in Metz, France. The objective of the performance evaluation is to perform a stress test to observe the behavior of the O-MI gateway – mainly in terms of response time – under heavy load. The open-source software Apache JMeter[6] is used to simulate the load on the O-MI gateway for the experiment. The requests are sent from the university network in Luxembourg from a MAC Book Pro Retina (mi-2015) with a CPU Intel Core i7 2.8GHz and the memory of 16GB 1600MHz DDR3.

The experiment is designed as follows. The simulated users send O-MI/O-DF requests to receive, in return, parking-related information generated by the O-MI gateway (cf. Figure 7.3). The number of concurrent users increases gradually (in groups of 10 users), as depicted in Figure 7.5d, up to 30 concurrent users request the same information. After 500 seconds, the number of users is decreased 10-by-10 until the end of the experiment (1000 seconds). Two load scenarios are considered: in the first one, users only request data about a single parking facility ('small' request of size 2605 bytes each), whereas in the second one the request is extended to the whole data

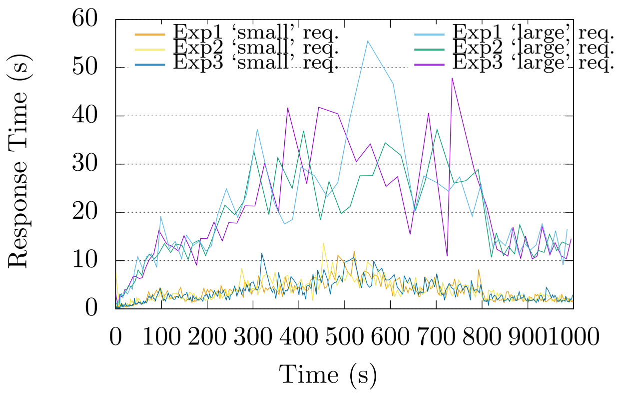
[6] http://jmeter.apache.org/



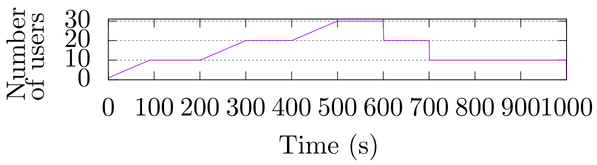
(a) Scenario 1: Response Time considering 'small' request load.



(b) Scenario 2: Response Time considering 'large' request load.



(c) User1's response time: Scenario 1 (small request) and Scenario 2 (large request).



(d) Evolution of the number of concurrent users.

Figure 7.5: Response time evolution (with concurrent users).

of parking facilities published by the O-MI node ('large' request of size 19740 bytes each). Each scenario is run only three times since the observed response times do not significantly evolve, as evidenced through Figure 7.5c in which the response time of the first simulated user is displayed. Figures 7.5a and 7.5b respectively provide an aggregated view of the results for both scenarios, as a boxplot provides the minimum, 1^{st} quartile, median, 3^{rd} quartile, and maximum of the response time for the three different user groups over each 100 second period (i.e., corresponding to the interval of time over which the number of users varies).

Based on the resulting response time boxplots of the first scenario (cf.Figure 7.5a), the following conclusions can be drawn:

- When the load is relatively low (i.e., between 1 and 10 concurrent users $[0s; 200s]$), the response time is about 2-4 seconds, which is already high for such Internet communications. This can be explained by the fact that (i) the O-MI gateway is hosted on a resource-constrained device (behind an Internet gateway with port redirection to be more precise) and (ii) each request/response needs five TCP segments in total (excluding the opening/closing connection and the acknowledgement frames).

- When the load increases (due to the increase of users $[200s; 600s]$), the response time increases accordingly. The response time can even reach more than 15 seconds depending on the synchronization of the concurrent requests on the O-MI server. It follows that the O-MI gateway cannot handle 30 users at the same time with low latency. However, the server is still capable to reply to all requests since the number of errors is very low, as shown in Table 7.2. Furthermore, the HTTP error code associated to all these errors is `400 Bad Request`, which implies that the errors occurred either at the sender or at the network level (e.g., an erroneous bit), but not at the server.

- Finally, when the number of users is reduced back to 10, the server progressively adapts itself and the response time decreases.

A similar conclusion can be drawn from the response time boxplots in Figure 7.5b (scenario 2). However, the response times are significantly higher (around 20-40 seconds for 30 concurrent users) due to the increased number of TCP segments (i.e., 16 segments excluding the opening/closing connection and acknowledgment) that are needed to generate and to transport all parking information. The maximum goes up to more than 150 seconds. In addition, the number of errors is substantially more important in scenario 2 ('large' request). As evidenced in Table 7.2, the O-MI node is not able to handle all the requests. The HTTP error codes associated to these errors are of type (i) `502 Bad Gateway` or (ii) `503 Service Unavailable`. It implies that (i) one gateway did not receive an answer from the server, or (ii) the service provided by the server is unavailable at that time. In both cases the server was unable to handle the request due to too many incoming requests at the same time.

Even for a smart mobility application, which does not require (hard) real-time data, this might become a serious problem for the development of applications. Thus, requests should be kept as small as possible. To this end, it is important that developers and/or connected Things can first discover one or more service items (e.g., only one parking item in Brussels) instead of requesting the whole data structure (e.g., all parking-related data) exposed by the O-MI gateway. The IoT service marketplace developed in bIoTope can eventually help developers to search for and access such

Table 7.2: Experiments Summary

Scenario	Exp	Group	No. of Req.	%Error
'Small' request (2605 bytes)	1	1	3115	0.35%
		2	242	0%
		3	117	0.41%
	2	1	2945	0.10%
		2	1166	0.09%
		3	241	0%
	3	1	3099	0.32%
		2	1171	0%
		3	255	0%
'Large' request (19740 bytes)	1	1	544	1.47%
		2	203	2,96%
		3	39	28,21%
	2	1	549	0.91%
		2	199	1.51%
		3	34	11.76%
	3	1	538	1.30%
		2	189	4.23%
		3	40	7.5%

service items depending on their location, service type and/or reputation, etc. Such an architecture – having resource-constrained devices at the edge of the network and a powerful server at the marketplace level – helps unload the incoming traffic at the O-MI gateway level, thus offering low response times. The effort is therefore drifted from the IoT data publishers to an IoT intermediary service, namely the IoT service marketplace.

In a second experiment, it was investigated whether the harmonized data formats (O-DF-based) impact the application performance compared to the direct access to the proprietary APIs. To this end, the parsing time of the two (proprietary) data sources – namely (i) Brussels-related data accessed from the open data portal of Brussels Region (formatted using XML), and (ii) Lyon-related data accessed from the open data portal of Grand Lyon (formatted using JSON) – is compared with the parsing of the harmonized O-DF structure (in XML) considering the two corresponding O-MI gateways (one exposing Brussels-related data and one exposing Lyon-related data, as depicted in Figure 7.2). The experiment is run with Java™, relying on the JAXB library for parsing XML and the native Java library to parse JSON objects. The experiment results are shown in Table 7.3 (considering only the time required to parse the string response into internal objects). It can be noted that the time for accessing the both non-standardized APIs is significantly higher to the one for accessing the O-MI gateway, even though this does not really impact on the quality of user experience. The reason for this time difference is that the data is already pre-processed when accessing the O-DF payload at the gateway level. However, compared to the overall latency in collecting the O-MI/O-DF messages (cf. previous experiment), the absolute values are not significant (ms against s). Nonetheless, one of the main benefits remains the reduced complexity and development effort for developers to understand and integrate

Table 7.3: Parsing Performance Comparison (in Java)

Approach	Payload (bytes)	Avg. (ms)	Std. dev. (ms)
Traditional	Overall: 24766 Brussels (XML): 10212 Lyon (JSON): 14554	100.74	2.96
O-DF	Overall (XML): 47537	34.42	2.04

heterogeneous data sources.

7.4 Discussion

To open the discussion section about the approach for enhanced interoperability proposed in this chapter, the links between the building blocks of the bIoTope ecosystem and LCIM are presented in Table 7.4, while highlighting which layers the productivity tool prototype directly and indirectly supports. The presented productivity tool directly contributes to semantic and pragmatic interoperability. Supporting the design of a standardized data structure by suggesting known semantic terms for annotation aims at semantic interoperability. The application of the tool to O-MI/O-DF addresses pragmatic interoperability, as O-MI and O-DF together form a service description, i.e. define how to read data objects and call methods.

Table 7.4: LCIM Mapped to bIoTope Building Blocks

Interoperability layers		bIoTope approach	Prod. Tool
6	Conceptual	*Open* policy	(✓)
5	Dynamic	IoT marketplace	(✓)
4	Pragmatic	O-MI	✓
3	Semantic	O-DF + ontologies	✓
2	Syntactic	XML	*compliant*
1	Technical	HTTP, etc.	*compliant*

This table reveals that our approach relies indirectly on existing interoperability mechanisms for the lower layers. The O-MI standard (used for the presented prototype) is mainly built upon the HTTP stack (technical interoperability) and uses XML as a serialization (syntactical interoperability). The presented productivity tool prototype does not contribute to these levels, but relies and complies with existing solutions of the bIoTope initiative. The indirect influence of the proposed productivity tool on dynamic and conceptual interoperability is more significant. This is because the discovery of data and services via the IoT marketplace (at the dynamic interoperability level) highly depends on the semantic annotations. At the same time, the bIoTope ecosystem follows an open vision that does not statically impose a conceptual model on all data/service providers and consumers, but rather aims to guide data/service providers to find the right model for their use case. Such a guidance can be achieved thanks to an approach (and associated productivity tools) like the one presented in this chapter.

The proposed productivity tool, as of now, relies on the LOV repository. If a certain term is not available in existing ontologies, the user is able to add custom elements to the data structure. This could be an advantage in terms of flexibility, however, it also leads to inconsistencies in the semantic model. In addition, the selection of the right ontology term(s) is not very intuitive from the ranking of terms returned by LOV's API. Users might have a different technical and domain-related background and might want to reuse ontology terms based on custom preferences. As discussed throughout this dissertation, an ORT specifically designed for this purpose could enhance this process. Furthermore, even though the tool aims at reducing the overhead for such a process, it is still challenging to motivate IoT data/service providers to publish their IoT resources based upon open, standardized APIs as long as the process is not fully automated. However, from an ecosystem perspective, such a productivity tool could also be applied by consumers themselves, e.g., by *system integrators* of companies who aim to provide all company departments with a harmonized way of accessing and understanding data from various and disparate information systems. Another limitation for the bIoTope prototype arises through the design of O-DF, which was not designed to represent RDF but rather for describing IoT resources in a simple manner. The integration of terms from RDF ontologies can be done through certain attributes of the O-DF standard that allow for the specification of URIs as values.

7.5 Conclusion

This chapter presents a productivity tool that allows to publish IoT data and services with minimal effort to the WoT. The steps introduced by the conceptual design of the tool include the development of the data structure based on semantic ontologies, mapping of proprietary formats and entities, and the generation of an IoT gateway agent that can be deployed on any devices at the edge of the WoT. The prototype and use case is implemented in the framework of the H2020 bIoTope project (part of the IoT-EPI initiative), whose resulting IoT gateway agents are deployed to expose smart city-related data (i.e., Grand Lyon and Brussels Region in the presented use case) through the adopted open standardized API named O-MI and O-DF.

In conclusion, if O-MI nodes are hosted on resource-constrained devices, the requests should be formulated as specific as possible. The discovery of relevant IoT data or services (referred to as *Objects* and *InfoItems* in O-DF) can be optimized through potential IoT search engines and associated service marketplaces (as the one investigated in bIoTope), which is designed as a scalable cloud service.

This chapter presents an application that integrates ontology recommendation in the context of an IoT ecosystem. In the next chapter, we conclude the dissertation and present future work.

Chapter 8

Conclusion

This chapter concludes the dissertation. We summarize the contributions, outline directions for future work and present concluding remarks as an outlook.

Contents

8.1 Summary of Contributions

According to Gartner, the number of *"IoT endpoints"* in the industry is expected to increase by 21% in 2020, compared with 2019, to reach 5.8 billion endpoints[1]. With more and more IoT endpoints, networks, vendors, and platforms, handling the heterogeneity of respectively generated data becomes an increasing challenge. In order to develop intelligent and pervasive systems that could improve everyday life in domains such as healthcare, transportation, production, environmental protection and assisted living, the ability for these system to discover, access and understand relevant data on demand is crucial. We refer to this issue as the semantic interoperability challenge.

One way to start moving towards a truly connected IoT setting is the establishment of an open data ecosystem. Conceptually, the IoT ecosystem provides incentives and regulations for data publication and consumption, e.g., through data trading via an IoT marketplace component. An open ecosystem with respective platforms is designed in a way that allows any actor to join by adapting to the respective technologies. The research community and industry alike actively explore the use of Semantic Web technologies to enable semantic interoperability in such IoT ecosystem settings. In Chapter 1, we identify four challenges that arise when applying Semantic Web technologies in order to achieve semantic interoperability in IoT ecosystems: the development of respective ontologies, ontology mappings, enabling ontology reuse and supporting the use of ontologies in IoT use cases. This highlights that semantic interoperability in the IoT is not simply achieved by enabling respective tools and platforms with Semantic Web technologies. The fundamental assumption and practical challenge rather lie in establishing a real-world, continuous consensus on the data models to be used within the ecosystem, so that both actors and systems are able to understand and reuse the data.

In this dissertation, we present a vision that proposes ontology recommendation as a key building block of IoT ecosystems in order to achieve semantic interoperability. Unlike administering a standardized set of data models in a top-down manner, relying on Semantic Web technologies, including open ontologies, allows data models to evolve dynamically following ecosystem actors' consensus. Under this vision, this dissertation investigates the applicability of existing ontology recommendation techniques for the IoT, and further addresses several identified shortcomings in the scope of ontology recommendation and the IoT ecosystem vision. The contributions are summarized in the following.

A taxonomy for ontology recommendation tools. Chapter 4 presents a thorough review and conceptualization of state-of-the-art in ontology recommendation and the consideration in IoT ecosystems. We identify several limitations and future directions for six key dimensions of ontology recommendation, namely ontology collection, evaluation, curation, tool interaction, matching and ranking. Among other gaps, this survey reveals that the evaluation of recommendation criteria is relatively unexplored. Also, many domain-dependent solutions exist, for which respective use cases fundamentally drive the design of the ORTs, however, we find that the application of ontology recommendation for IoT use cases has not been investigated in the literature.

[1] `https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io`

Efficient ranking for IoT ontology collections. In Chapter 5, we highlight and address key issues in state-of-the-art ontology ranking approaches: the main criteria for ontology reuse, *popularity*, in its form as commonly used in existing tools, is not efficient for IoT ontology collections. This is caused by the currently missing feedback of ontology usages in real-world IoT projects. We address this issue by proposing a novel approach to ontology ranking, which integrates known popularity measures in the learning target of a ranking model trained using Learning To Rank (LTR). We show that incorporating qualitative attributes that can be directly extracted from the ontology (i.e., without relying on external metadata) helps to improve the efficiency of ontology ranking models and are thus more suitable to be applied for ORTs targeting the IoT domain.

Domain-independent ontology ranking benchmark. Ontology ranking forms one of the most important aspects of ontology recommendation, since the ranking typically incorporates many recommendation criteria. However, we identify several limitations in the state-of-the-art of evaluating the ranking models of existing ORTs. Ranking evaluation in this context is a challenging task, mainly due to the different ontology repositories used and the complex issue of defining relevance labels. In an effort to move towards a domain-independent, large-scale ontology ranking benchmark, we propose LOVBench in Chapter 6. LOVBench exploits real-world user click logs collected from the LOV platform to derive relevance labels. LOVBench outperforms the state-of-the-art in terms of granularity (classes and properties), number of queries and judgments, and number of features considered for the ranking comparisons. Our analysis further allows to observe real-world user behavior and our findings dissent dominant assumptions in the literature, e.g., that users actually do not often use multiple words for their keyword query and that users equally look for classes and properties. We propose adapted ranking features that take these considerations into account and outperform the state-of-the-art.

Ontology recommendation integration in an IoT ecosystem application. Finally, Chapter 7 presents an application in an IoT ecosystem context that benefits from ontology recommendation. We propose a methodology to efficiently map data from IoT gateways to semantically annotated, standardized ones, fundamentally relying on ontology term recommendations. We apply this approach to a smart parking use case and adopt the IoT ecosystem technology stack developed in the bIoTope project. The use case considers data from different cities (Lyon and Brussels) to emphasize the interoperability issue. Furthermore, we evaluate the performance of the respective gateways based on several load scenarios.

8.2 Directions for Future Work

This section describes future research directions and potential extensions of the presented work.

Towards a unified, real-world ontology ranking benchmark. With LOVBench, we propose the first ontology ranking benchmark that is applicable to real-world ontology term ranking evaluation. Nevertheless, it is limited to the user feedback that could be collected from the LOV platform. By adopting the LOVBench to click logs from other ORTs and integrating these in the benchmark, it could become a more general, representative ground truth for evaluation. This, potentially, could extend LOVBench also with other query formats. Lastly, we kept LOVBench purposely domain-independent – adopting LOVBench with domain-specific ranking features would further allow novel insights in the impact of ontology ranking features. Since, to the best of our knowledge, user clicks have not been used for the evaluation of ontology ranking models before, an adoption of the LOVBench approach to other ORTs could eventually result in a 'landscape' of benchmarks, applicable to all variations of respective selection dimensions.

Improved ranking evaluation through online experiments. The ultimate test of a ranking model is to deploy it and to observe the user behavior. E.g., if users in average perform clicks at a higher position, it is an indicator that the ranking model performs better in ranking the most relevant documents at the top. Both experiments in Chapter 5 and 6 have been performed offline on a previously collected ground truth. To strengthen the evaluation and to confirm the validity of the insights, these ranking models could be deployed in an existing ORT and compared to their predecessors with actual user feedback. Such online experiments have previously been conducted for ranking evaluation purposes (cf. [156]).

Combination of ontologies and terms. The contributions of this dissertation with regard to ontology ranking are focused on ranking single ontologies and terms. However, the IoT is a great example in which many terms of different ontologies from various domains need to be combined. To the best of our knowledge, it is not at all explored how combinations of ontologies and terms should be recommended. Should higher coverage or higher ontology quality be preferred? Should the number of ontologies be minimal when recommending term combinations? Which input types are most useful in common use cases, and particularly in the IoT? An ideal solution for the IoT would be to take a conventional reading from an IoT gateway with a custom data model (e.g., in the form of JSON or XML) and provide a complete and automated mapping to ontological terms. Future work could follow this direction, starting from the ranking and its evaluation of ontology/term combinations. Existing work that provides ranking for combinations of ontologies includes NCBO 2.0 [141], yet lacking a thorough evaluation and term-level recommendations.

Understanding user intent and information need. The studies in this dissertation are further limited by the circumstance that they are evaluated based on implicit user feedback: the intent and actual information need are unknown. However, user intent and behavior in the context of ontology reuse has been investigated in the literature before (cf. [196]). Future work, nevertheless, should include further user studies, especially in the context of IoT, to understand how ontology recommendation can be best designed and integrated in tools to support users.

Ontology search integration in productivity tools. With PROFICIENT, we solely demonstrate with a prototype the feasibility of integrating ontology recommendation to support IoT use cases. However, state-of-the-art tools that support, e.g., ontology mapping and data transformation are much richer in functionality.

The majority of these tools assume that the targeted schema is known, such as for the *named entity recognition* task that aims to find the best formally specified concepts (from a given knowledge representation) from natural language text. Given that neither the data nor the schema are static in IoT settings, it may not be appropriate to consider the development of a target schema and the mapping from a source to a target schema as independent. We believe future research could further explore the conceptual integration of ontology recommendation in such tasks.

8.3 Concluding Remarks

The realization of IoT ecosystems faces various obstacles and, from a technical perspective, this dissertation contributes to ontology recommendation approaches that can facilitate this vision and enable the provision of efficient ontology recommendation for actors in such ecosystems. We can observe that industrial and academic research initiatives are developing future IoT platforms and, in this scope, ontology recommendation allows the establishment of community consensus on common data models for the IoT.

However, the current development can still be seen critical. As stated in the motivation of this dissertation, the sole use of ontologies does not yet establish cross-platform interoperability. As long as *semantics of data* and *design of software/services* are not strictly independent from each other, and as long as key ecosystem building blocks, such as the IoT market place, have semantics hard-coded in their software, we are essentially adding new vertical silos to existing set of vertical silos. Prof. James Hendler's famous quote "*a little semantics goes a long way*" of course remains true – with the platforms already relying on Semantic Web technologies, it certainly becomes easier to integrate data from different platforms, and the W3C WoT Thing Descriptions is a good example that will foster the important high-level understanding of different data sources. However, this approach does not at all prevent a scenario in which a few IoT platforms will emerge and simply dominate the markets (establishing their standards and models) based on the huge amount of data that is controlled through these platforms. We can easily disclose similarities with today's social media, in which user's data is owned by large, market-dominant companies. *Decentralized Web* movements aim to tackle the issues that arise from this situation and are gaining more traction. The Social Linked Data (SOLID) project[2] lead by Tim Berners-Lee, e.g., aims to move data ownership from platforms (as we can see today) to users – promoting an approach that is solely built on existing Web technologies, including the Semantic Web. The IoT is at the risk of facing the similar fate as social media by being dominated by a few large IoT service providers. Such circumstance will, ultimately, limit interoperability, innovation, and the value creation that could be ideally achieved. The vision of a decentralized Web thus applies similarly well to the IoT, however, this development is out of the scope on technical issues considered in this dissertation. From a technology perspective, however, the decentralized Web still faces semantic interoperability issues. Instead of relying on a single standard, the goal is to support different data models that represent *local truths* and future software and services may need to be able to cope with *partial understanding* of data.

Independent from the way in which the IoT and future Web data ecosystems will

[2] `https://solid.mit.edu/`

develop, we believe that it will always be a fundamental task to find and reuse existing data models. The Semantic Web and its public ontologies provide a perfect infrastructure and starting point to achieve this, and may in the long run help to reach a higher degree of interoperability among computing systems. However, the IoT ecosystem vision and ontology reuse in general also faces various non-technical obstacles. For a variety of reasons, ecosystem actors may simply not be interested in ontology reuse. Examples include actors who want to remain in control over their vocabulary, want to add a 'novel' approach to their portfolio, and purposely *not* to interoperate with certain other ecosystem actors and systems. The designs of forthcoming IoT ecosystems need to provide the right incentives to make actors interested in ontology reuse.

Lastly, it is worth mentioning that the use of (IoT) data, metadata and development of services as of today varies a lot in sciences and industry. It is reasonable to believe that achieving interoperability in industrial settings is much more difficult than in the sciences, and would require an overarching body that promotes and negotiates ecosystem principles, such as FAIR. The Semantic Web with efficient ORTs in place for respective domains may foster and support the convergence to commonly accepted principles.

Appendices

Chapter A

List of Publications, Resources and Research Activities

Papers included in the dissertation:

- 2020
  - Niklas Kolbe, Pierre-Yves Vandenbussche, Sylvain Kubler, and Yves Le Traon. "LOVBench: Ontology Ranking Benchmark". In: *Proceedings of The Web Conference (WWW)*. ACM, 2020. DOI: `10.1145/3366423.3380245`
- 2019
  - Niklas Kolbe, Sylvain Kubler, and Yves Le Traon. "Popularity-driven Ontology Ranking using Qualitative Features". In: *International Semantic Web Conference*. Springer. 2019. DOI: `10.1007/978-3-030-30793-6_19`
  - Niklas Kolbe, Sylvain Kubler, Jérémy Robert, Yves Le Traon, and Arkady Zaslavsky. "Linked Vocabulary Recommendation Tools for Internet of Things: A Survey". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 127. DOI: `10.1145/3284316`
- 2017
  - Niklas Kolbe, Jérémy Robert, Sylvain Kubler, and Yves Le Traon. "PROFICIENT: Productivity Tool for Semantic Interoperability in an Open IoT Ecosystem". In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*. ACM. 2017. DOI: `10.1145/3144457.3144479`

Papers not included in the dissertation:

- 2017
  - Jérémy Robert, Sylvain Kubler, Niklas Kolbe, Alessandro Cerioni, Emmanuel Gastaud, and Kary Främling. "Open IoT Ecosystem for Enhanced Interoperability in Smart Cities–Example of Métropole De Lyon". In: *Sensors* 17.12 (2017), p. 2849. DOI: `10.3390/s17122849`

- Niklas Kolbe, Arkady Zaslavsky, Sylvain Kubler, Jérémy Robert, and Yves Le Traon. "Enriching a Situation Awareness Framework for IoT with Knowledge Base and Reasoning Components". In: *International and Interdisciplinary Conference on Modeling and Using Context*. Springer. 2017, pp. 41–54. DOI: 10.1007/978-3-319-57837-8_4
- Niklas Kolbe, Sylvain Kubler, Jérémy Robert, Yves Le Traon, and Arkady Zaslavsky. "Towards Semantic Interoperability in an Open IoT Ecosystem for Connected Vehicle Services". In: *Global Internet of Things Summit (GIoTS)*. IEEE. 2017. DOI: 10.1109/GIOTS.2017.8016229

Resources:

- LOVBench: a ground truth for ontology ranking, LTR datasets for an empirical comparison of ranking models, and analysis of real-world user click feedback from the LOV platform.
  https://github.com/nut-hatch/LOVBench
- IoT Ontology Ranking Experiments: source files and resources for LTR experiments to compare ranking models applied to an IoT ontolgoy collection.
  https://github.com/nut-hatch/OntologyRankingExperiments

Other research activities:

- EU H2020 IoT-EPI research project member: *bIoTope*
  https://biotope-project.eu/
- Workshop participation: *Dagstuhl Workshop on Interoperability of Metadata Standards in Cross- Domain Science, Health, and Social Science Applications*
  https://ddi-alliance.atlassian.net/wiki/spaces/DDI4/pages/684195863/2019+Workshop+on+Interoperability+of+Metadata+Standards+in+Cross-Domain+Science+Health+and+Social+Science+Applications+II
- Journal review: *Journal of Systems and Software (JSS)*
  https://www.journals.elsevier.com/journal-of-systems-and-software

Chapter B

List of Ontologies and Terms

In Table B.1 we provide an overview of all ontologies and terms that are referred to in this dissertation.

Table B.1: Overview of Ontologies and Terms

Prefix	Ontology and Terms
`rdf`	`http://www.w3.org/1999/02/22-rdf-syntax-ns#`
The fundamental RDF concepts.	
	`rdf:Property`
	`rdf:type`
`rdfs`	`http://www.w3.org/2000/01/rdf-schema#`
Basic vocabulary for RDF.	
	`rdfs:Class`
	`rdfs:subClassOf`
	`rdfs:label`
	`rdfs:comment`
	`rdfs:`
	`rdfs:domain`
	`rdfs:range`
`owl`	`http://www.w3.org/2002/07/owl#`
Extended vocabulary for RDF, complements RDFS.	
	`owl:Class`
	`owl:ObjectProperty`
	`owl:DatatypeProperty`
	`owl:AnnotationProperty`
	`owl:OntologyProperty`
	`owl:imports`
`schema`	`http://schema.org/`
Vocabulary maintained and used by major search engines.	
	`schema:CivicStructure`
	`schema:ParkingFacility`
	`schema:Vehicle`
	`schema:openingHours`

Table B.1: Overview of Ontology and Terms (Cont.)

Prefix	Ontology and Terms
	`schema:geo`
	`schema:latitude`
	`schema:longitude`
`sosa`	`http://www.w3.org/ns/sosa#`
Describes the concepts Sensor, Observation, Sample, and Actuator (SOSA).	
	`sosa:FeatureOfInterest`
	`sosa:Platform`
	`sosa:Observation`
	`sosa:hasFeatureOfInterest`
	`sosa:hasSimpleResult`
	`sosa:resultTime`
	`sosa:observedProperty`
`mobivoc`	`http://schema.mobivoc.org/#`
The Open Mobility Vocabulary (MobiVoc).	
	`mobivoc:ParkingFacility`
	`mobivoc:ParkingFacilityEntrance`
	`mobivoc:Car`
	`mobivoc:entrance`
	`mobivoc:rateOfOccupancy`
	`mobivoc:capacity`
	`mobivoc:validForVehicle`
	`mobivoc:pedestrianAccess`
	`mobivoc:maximumValue`
`dcterms`	`http://purl.org/dc/terms/`
Metadata terms defined by the Dublin Core Metadata Initiative (DCMI).	
	`dcterms:creator`
	`dcterms:title`
	`dcterms:description`
`dce`	`http://purl.org/dc/elements/1.1/`
Metadata Element Set maintained by DCMI.	
	`dce:title`
	`dce:description`
`skos`	`http://www.w3.org/2004/02/skos/core`
A common data model for sharing and linking knowledge organization systems.	
	`skos:prefLabel`
	`skos:altLabel`
`m3`	`http://sensormeasurement.appspot.com/m3#`
Ontology that describes types of sensors, measurements and units.	
	`m3:hasContext`
	`m3:hasM2MDevice`

Chapter C

List of Symbols

A summary of the most commonly used variables in this dissertation is provided in Table C.1.

Table C.1: Notations

Symbol	Meaning
R	Ontology repository, i.e., a set of ontologies O
O	Ontology, i.e., a set of RDF triples `(subject, predicate, object)`
t	Term defined in an ontology
c	A term of type class
p	A term of type property
Q	A keyword user query
q_i	ith word in query Q
D	Dictionary, a set of words
w	A word
$\sigma_{\mathcal{T}}$	Set of terms in ontology O that match query Q, where $\mathcal{T}$ corresponds to the type of matched terms, i.e., $\mathcal{T} \in \{t, c, p\}$ (corresponding to all terms, only classes, or only properties, respectively)
$G(O)$	Ontology graph of O
$G(R, \mathcal{P})$	Repository graph of R considering the ontology relations in $\mathcal{P}$
$Rel_{Q,t}$	Relevance for a query-term pair
S_t	User's satisfaction probability
C_t	User's click probability
E_t	User's examination probability
M	A model
$\mathcal{S}$	A set of search sessions (e.g., from a training set)
s	A search session
Φ	Ranking scoring function

References

[1] Alberto Abella, Marta Ortiz-de Urbina-Criado, and Carmen De-Pablos-Heredero. "A model for the analysis of data-driven innovation and value generation in smart cities' ecosystems". In: *Cities* 64 (2017), pp. 47–53. DOI: `10.1016/j.cities.2017.01.011` (cit. on p. 4).

[2] Aman Agarwal, Xuanhui Wang, Cheng Li, Michael Bendersky, and Marc Najork. "Addressing Trust Bias for Unbiased Learning-to-Rank". In: *The World Wide Web Conference*. ACM. 2019, pp. 4–14. DOI: `10.1145/3308558.3313697` (cit. on p. 87).

[3] Bengt Ahlgren, Markus Hidell, and Edith C-H Ngai. "Internet of Things for Smart Cities: Interoperability and Open Data". In: *IEEE Internet Computing* 20.6 (2016), pp. 52–56. DOI: `10.1109/MIC.2016.124` (cit. on pp. 4, 6).

[4] Bram van den Akker, Ilya Markov, and Maarten de Rijke. "ViTOR: Learning to Rank Webpages Based on Visual Features". In: *The World Wide Web Conference*. ACM. 2019, pp. 3279–3285. DOI: `10.1145/3308558.3313419` (cit. on pp. 87, 97).

[5] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2347–2376. DOI: `10.1109/COMST.2015.2444095` (cit. on p. 8).

[6] Harith Alani, Christopher Brewster, and Nigel Shadbolt. "Ranking Ontologies with AKTiveRank". In: *International Semantic Web Conference*. Springer. 2006, pp. 1–15. DOI: `10.1007/11926078_1` (cit. on pp. 38, 41, 51, 59, 84, 86, 88, 89, 94–99).

[7] Darko Andročec, Matija Novak, and Dijana Oreški. "Using Semantic Web for Internet of Things Interoperability: A Systematic Review". In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 14.4 (2018), pp. 147–171. DOI: `10.4018/IJSWIS.2018100108` (cit. on pp. 7, 72, 76).

[8] Grigoris Antoniou and Frank Van Harmelen. "Web Ontology Language: Owl". In: *Handbook on Ontologies*. Springer, 2004, pp. 67–92. DOI: `10.1007/978-3-540-24750-0_4` (cit. on pp. 6, 14).

[9] Jamshaid Ashraf, Omar Khadeer Hussain, and Farookh Khadeer Hussain. "Making sense from Big RDF Data: OUSAF for measuring ontology usage". In: *Software: Practice and Experience* 45.8 (2015), pp. 1051–1071. DOI: `10.1002/spe.2272` (cit. on pp. 37, 57).

[10] Ghislain Auguste Atemezing and Raphaël Troncy. "Information content based ranking metric for linked open vocabularies". In: *Proceedings of the 10th International Conference on Semantic Systems*. ACM. 2014, pp. 53–56. DOI: `10.1145/2660517.2660533` (cit. on p. 51).

[11] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey". In: *Computer networks* 54.15 (2010), pp. 2787–2805. DOI: `10.1016/j.comnet.2010.05.010` (cit. on p. 8).

[12] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. "LODStats – An Extensible Framework for High-performance Dataset Analytics". In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2012, pp. 353–362. DOI: `10.1007/978-3-642-33876-2_31` (cit. on pp. 36, 57).

[13] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN: 020139829X (cit. on pp. 25, 99).

[14] Garvita Bajaj, Rachit Agarwal, Pushpendra Singh, Nikolaos Georgantas, and Valerie Issarny. "A study of existing Ontologies in the IoT-domain". In: (2017) (cit. on pp. 8, 65).

[15] Noorani Bakerally, Olivier Boissier, and Antoine Zimmermann. "Smart city artifacts web portal". In: *International Semantic Web Conference*. Springer. 2016, pp. 172–177. DOI: `10.1007/978-3-319-47602-5\_34` (cit. on pp. 8, 22).

[16] Debasis Bandyopadhyay and Jaydip Sen. "Internet of things: Applications and challenges in technology and standardization". In: *Wireless personal communications* 58.1 (2011), pp. 49–69. DOI: `10.1007/s11277-011-0288-5` (cit. on p. 9).

[17] Payam Barnaghi, Wei Wang, Lijun Dong, and Chonggang Wang. "A Linked-data Model for Semantic Sensor Streams". In: *IEEE International Conference on Internet of Things (iThings/CPSCom), IEEE Green Computing and Communications (GreenCom), IEEE Cyber, Physical, and Social Computing*. IEEE. 2013, pp. 468–475. DOI: `10.1109/GreenCom-iThings-CPSCom.2013.95` (cit. on p. 66).

[18] Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. "Semantics for the Internet of Things: early progress and back to the future". In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 8.1 (2012), pp. 1–21. DOI: `10.4018/jswis.2012010101` (cit. on pp. 6–8, 65, 108).

[19] Alexandros Batzios and Pericles A Mitkas. "WebOWL: A Semantic Web search engine development experiment". In: *Expert Systems with Applications* 39.5 (2012), pp. 5052–5060. DOI: `10.1016/j.eswa.2011.11.034` (cit. on pp. 36, 57).

[20] Martin Bauer, Hamza Baqa, Sonia Bilbao, Aitor Corchero, Laura Daniele, Iker Esnaola, et al. *Towards semantic interoperability standards based on ontologies*. Tech. rep. 2019 (cit. on p. 4).

[21] Hamid R Bazoobandi, Jacopo Urbani, Frank van Harmelen, and Henri Bal. "An Empirical Study on How the Distribution of Ontologies Affects Reasoning on the Web". In: *International Semantic Web Conference*. Springer. 2017, pp. 69–86. DOI: 10.1007/978-3-319-68288-4_5 (cit. on p. 68).

[22] Tim Berners-Lee. *Linked Data: Design Issues*. 2006. URL: http://www.w3.org/DesignIssues/LinkedData.html (cit. on pp. 6, 49, 50).

[23] Tim Berners-Lee, James Hendler, Ora Lassila, et al. "The Semantic Web". In: *Scientific American* 284.5 (2001), pp. 28–37 (cit. on pp. 6, 14).

[24] Diego Berrueta, Jon Phipps, Alistair Miles, Thomas Baker, and Ralph Swick. "Best practice recipes for publishing RDF vocabularies". In: *Working draft, W3C* (2008) (cit. on pp. 6, 49).

[25] Christian Bizer, Tom Heath, and Tim Berners-Lee. "Linked Data: The Story So Far". In: *Semantic Services, Interoperability and Web Applications: Emerging Concepts*. IGI Global, 2011, pp. 205–227. DOI: 10.4018/978-1-60960-593-3.ch008 (cit. on pp. 15, 84).

[26] E Paslaru Bontas, Malgorzata Mochol, and Robert Tolksdorf. "Case studies on ontology reuse". In: *Proceedings of the IKNOW05 International Conference on Knowledge Management*. Vol. 74. 2005, pp. 345–353 (cit. on p. 19).

[27] Janez Brank, Marko Grobelnik, and Dunja Mladenić. "A Survey of Ontology Evaluation Techniques". In: *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)* (cit. on p. 49).

[28] D Brock and E Schuster. "On the semantic web of things". In: *Semantic Days* (2006), pp. 26–27 (cit. on p. 6).

[29] Arne Bröring, Stefan Schmid, Corina-Kim Schindhelm, Abdelmajid Khelil, Sebastian Käbisch, Denis Kramer, et al. "Enabling IoT Ecosystems through Platform Interoperability". In: *IEEE Software* 34.1 (2017), pp. 54–61. DOI: 10.1109/MS.2017.2 (cit. on p. 4).

[30] Arne Bröring, Andreas Ziller, Victor Charpenay, Aparna S Thuluva, Darko Anicic, Stefan Schmid, et al. "The BIG IoT API-Semantically Enabling IoT Interoperability". In: *IEEE Pervasive Computing* 17.4 (2018), pp. 41–51. DOI: 10.1109/MPRV.2018.2873566 (cit. on p. 7).

[31] Paul Buitelaar, Thomas Eigner, and Thierry Declerck. "OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection". In: *Proceedings of the Demo Session at the International Semantic Web Conference*. 2004 (cit. on pp. 30, 50, 57).

[32] Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, et al. "Learning to Rank using Gradient Descent". In: *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 89–96. DOI: 10.1145/1102351.1102363 (cit. on p. 99).

[33] Christopher JC Burges. "From ranknet to lambdarank to lambdamart: An overview". In: *Learning* 11.23-581 (2010), p. 81 (cit. on p. 77).

[34] Andrew Burton-Jones, Veda C Storey, Vijayan Sugumaran, and Punit Ahluwalia. "A semiotic metrics suite for assessing the quality of ontologies". In: *Data & Knowledge Engineering* 55.1 (2005), pp. 84–102. DOI: 10.1016/j.datak.2004.11.010 (cit. on pp. 31, 57).

[35] Fabian Burzlaff, Nils Wilken, Christian Bartelt, and Heiner Stuckenschmidt. "Semantic Interoperability Methods for Smart Service Systems: A Survey". In: *IEEE Transactions on Engineering Management* (2019). DOI: 10.1109/TEM.2019.2922103 (cit. on p. 9).

[36] Anila Sahar Butt, Armin Haller, and Lexing Xie. "A Taxonomy of Semantic Web Data Retrieval Techniques". In: *Proceedings of the 8th International Conference on Knowledge Capture*. ACM, 2015, pp. 1–9 (cit. on p. 30).

[37] Anila Sahar Butt, Armin Haller, and Lexing Xie. "DWRank: Learning Concept Ranking for Ontology Search". In: *Semantic Web* 7.4 (2016), pp. 447–461. DOI: 10.3233/SW-150185 (cit. on pp. 39, 41, 58, 77, 84, 88, 89, 94, 96, 98, 99).

[38] Anila Sahar Butt, Armin Haller, and Lexing Xie. "Ontology Search: An Empirical Evaluation". In: *International Semantic Web Conference*. Springer. 2014, pp. 130–147. DOI: 10.1007/978-3-319-11915-1_9 (cit. on pp. 21, 41, 53, 63, 84, 86–89, 91, 93–100).

[39] Anila Sahar Butt, Armin Haller, and Lexing Xie. "RecoOn: Ontology Recommendation for Structureless Queries". In: *Applied Ontology* 11.4 (2016), pp. 301–324. DOI: 10.3233/AO-160173 (cit. on pp. 40, 59).

[40] Anila Sahar Butt, Armin Haller, and Lexing Xie. "Relationship-based Top-K Concept Retrieval for Ontology Search". In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2014, pp. 485–502. DOI: 10.1007/978-3-319-13704-9_37 (cit. on pp. 39, 51, 58).

[41] Iván Cantador, Miriam Fernández Sánchez, and Pablo Castells. "Improving ontology Recommendation and Reuse in WebCORE by Collaborative Assessments". In: *Workshop on Social and Collaborative Construction of Structured Knowledge at 16th International World Wide Web Conference*. CEUR. 2007 (cit. on pp. 33, 58).

[42] Claudio Carpineto and Giovanni Romano. "A Survey of Automatic Query Expansion in Information Retrieval". In: *ACM Computing Surveys (CSUR)* 44.1 (2012), p. 1. DOI: 10.1145/2071389.2071390 (cit. on p. 52).

[43] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. "Expected reciprocal rank for graded relevance". In: *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM. 2009, pp. 621–630. DOI: 10.1145/1645953.1646033 (cit. on pp. 26, 27, 77).

[44] Olivier Chapelle and Ya Zhang. "A Dynamic Bayesian Network Click Model for Web Search Ranking". In: *Proceedings of the 18th International Conference on World Wide Web*. ACM. 2009, pp. 1–10. DOI: 10.1145/1526709.1526711 (cit. on p. 90).

[45] Gong Cheng, Saisai Gong, and Yuzhong Qu. "An Empirical Study of Vocabulary Relatedness and its Application to Recommender Systems". In: *The Semantic Web – ISWC 2011* (2011), pp. 98–113. DOI: 10.1007/978-3-642-25073-6_7 (cit. on p. 54).

[46] Gong Cheng and Yuzhong Qu. “Searching Linked Objects with Falcons: Approach, Implementation and Evaluation”. In: *International Journal on Semantic Web and Information Systems* 5.3 (2009), pp. 50–71. DOI: 10.4018/jswis.2009081903 (cit. on pp. 34, 57).

[47] Henry Chesbrough. “The Future of Open Innovation: The future of open innovation is more extensive, more collaborative, and more engaged with a wider variety of participants”. In: *Research-Technology Management* 60.1 (2017), pp. 35–38. DOI: 10.1080/08956308.2017.1255054 (cit. on p. 4).

[48] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. “Click Models for Web Search”. In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 7.3 (2015), pp. 1–115. DOI: 10.2200/S00654ED1V01Y201507ICR043 (cit. on pp. 25, 85, 90, 91).

[49] Andrea Cimmino, María Poveda-Villalón, and Raúl García-Castro. “eWoT: A Semantic Interoperability Approach for Heterogeneous IoT Ecosystems Based on the Web of Things”. In: *Sensors* 20.3 (2020), p. 822. DOI: 10.3390/s20030822 (cit. on p. 8).

[50] Mihai Codescu, Eugen Kuksa, Oliver Kutz, Till Mossakowski, and Fabian Neuhaus. “Ontohub: A semantic repository engine for heterogeneous ontologies”. In: *Applied Ontology* 12.3-4 (2017), pp. 275–298. DOI: 10.3233/AO-170190 (cit. on pp. 38, 58).

[51] Kevyn Collins-Thompson, Paul Bennett, Fernando Diaz, Charlie Clarke, and Ellen M Voorhees. “TREC 2013 Web Track Overview”. In: *TREC* (2013) (cit. on p. 97).

[52] Kevyn Collins-Thompson, Craig Macdonald, Paul Bennett, Fernando Diaz, and Ellen M Voorhees. “TREC 2014 Web Track Overview”. In: *TREC* (2015) (cit. on p. 97).

[53] Michael Compton, Payam Barnaghi, Luis Bermudez, et al. “The SSN ontology of the W3C semantic sensor network incubator group”. In: *Web semantics: science, services and agents on the World Wide Web* 17 (2012), pp. 25–32. DOI: 10.1016/j.websem.2012.05.003 (cit. on p. 7).

[54] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. “An Experimental Comparison of Click Position-Bias Models”. In: *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM. 2008, pp. 87–94. DOI: 10.1145/1341531.1341545 (cit. on pp. 26, 90).

[55] Li Da Xu, Wu He, and Shancang Li. “Internet of Things in Industries: A Survey”. In: *IEEE Transactions on Industrial Informatics* 10.4 (2014), pp. 2233–2243. DOI: 10.1109/TII.2014.2300753 (cit. on p. 9).

[56] Laura Daniele, Frank den Hartog, and Jasper Roes. “Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology”. In: *International Workshop Formal Ontologies Meet Industries*. Springer. 2015, pp. 100–112. DOI: 10.1007/978-3-319-21545-7_9 (cit. on p. 7).

[57] Mathieu d’Aquin, Claudio Baldassarre, Laurian Gridinoc, Marta Sabou, Sofia Angeletou, and Enrico Motta. “Watson: Supporting Next Generation Semantic Web Applications”. In: *Proceedings of the IADIS International Conference on WWW/Internet*. 2007 (cit. on pp. 34, 57).

[58] Mathieu d'Aquin and Holger Lewen. "Cupboard – A Place to Expose Your Ontologies to Applications and the Community". In: Springer, 2009, pp. 913–918. DOI: 10.1007/978-3-642-02121-3_81 (cit. on pp. 35, 58).

[59] Mathieu d'Aquin and Enrico Motta. "Watson, more than a Semantic Web search engine". In: *Semantic Web* 2.1 (2011), pp. 55–63. DOI: 10.3233/SW-2011-0031 (cit. on p. 34).

[60] Mathieu d'Aquin and Natalya F Noy. "Where to publish and find ontologies? A survey of ontology libraries". In: *Journal of Web Semantics* 11 (2012), pp. 96–111. DOI: 10.1016/j.websem.2011.08.005 (cit. on pp. 8, 21, 30, 46–48, 84, 87, 92).

[61] Giuseppe De Giacomo and Maurizio Lenzerini. "TBox and ABox Reasoning in Expressive Description Logics". In: *In Proceedings of KR-96* 96.316-327 (1996), p. 10 (cit. on p. 15).

[62] Daniele DellAglio, Emanuele Della Valle, Frank van Harmelen, and Abraham Bernstein. "Stream reasoning: A survey and outlook". In: *Data Science* 1.1-2 (2017), pp. 59–83. DOI: 10.3233/DS-170006 (cit. on pp. 8, 17).

[63] Hasan Derhamy, Jens Eliasson, and Jerker Delsing. "IoT Interoperability–On-Demand and Low Latency Transparent Multiprotocol Translator". In: *IEEE Internet of Things Journal* (2017). DOI: 10.1109/JIOT.2017.2697718 (cit. on p. 106).

[64] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R Scott Cost, Yun Peng, et al. "Swoogle: A Search and Metadata Engine for the Semantic Web". In: *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*. ACM. 2004, pp. 652–659. DOI: 10.1145/1031171.1031289 (cit. on pp. 21, 30, 41, 51, 57).

[65] Li Ding, Rong Pan, Tim Finin, Anupam Joshi, Yun Peng, and Pranam Kolari. "Finding and Ranking Knowledge on the Semantic Web". In: *International Semantic Web Conference*. Springer. 2005, pp. 156–170. DOI: 10.1007/11574620_14 (cit. on pp. 30, 41).

[66] Georges E Dupret and Benjamin Piwowarski. "A User Browsing Model to Predict Search Engine Click Data from Past Observations". In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 2008, pp. 331–338. DOI: 10.1145/1390334.1390392 (cit. on p. 90).

[67] Astrid Duque-Ramos, Jesualdo Tomás Fernández-Breis, Miguela Iniesta, Michel Dumontier, Mikel Egaña Aranguren, Stefan Schulz, et al. "Evaluation of the OQuaRE framework for ontology quality". In: *Expert Systems with Applications* 40.7 (2013), pp. 2696–2703. DOI: doi.org/10.1016/j.eswa.2012.11.004 (cit. on pp. 49–51).

[68] Orri Erling and Ivan Mikhailov. "RDF Support in the Virtuoso DBMS". In: *Networked Knowledge-Networked Media*. Springer, 2009, pp. 7–24. DOI: 10.1007/978-3-642-02184-8_2 (cit. on p. 16).

[69] Ivan Ermilov, Jens Lehmann, Michael Martin, and Sören Auer. "LODStats: The Data Web Census Dataset". In: *International Semantic Web Conference*. Springer. 2016, pp. 38–46. DOI: 10.1007/978-3-319-46547-0_5 (cit. on p. 36).

[70] Paola Espinoza-Arias, María Poveda-Villalón, Raúl García-Castro, and Oscar Corcho. "Ontological Representation of Smart City Data: From Devices to Cities". In: *Applied Sciences* 9.1 (2019), p. 32. DOI: 10.3390/app9010032 (cit. on pp. 72, 76).

[71] Yasmin Fathy, Payam Barnaghi, and Rahim Tafazolli. "Large-Scale Indexing, Discovery, and Ranking for the Internet of Things (IoT)". In: *ACM Computing Surveys (CSUR)* 51.2 (2018), p. 29. DOI: 10.1145/3154525 (cit. on pp. 9, 22).

[72] Miriam Fernández, Iván Cantador, and Pablo Castells. "CORE: A Tool for Collaborative Ontology Reuse and Evaluation". In: *4th International Workshop on Evaluation of Ontologies for the Web* (2006) (cit. on pp. 33, 58).

[73] Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski, Paweł Szmeja, and Katarzyna Wasielewska. "Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective". In: *Journal of Network and Computer Applications* 81 (2017), pp. 111–124. DOI: 10.1016/j.jnca.2016.08.007 (cit. on pp. 7, 65).

[74] Maria Ganzha, Marcin Paprzycki, Wiesław Pawłowski, Paweł Szmeja, Katarzyna Wasielewska, and Giancarlo Fortino. "Tools for Ontology Matching–Practical Considerations from INTER-IoT Perspective". In: *International Conference on Internet and Distributed Computing Systems*. Springer. 2016, pp. 296–307. DOI: 10.1007/978-3-319-45940-0_27 (cit. on p. 8).

[75] Birte Glimm, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. "Conjunctive Query Answering for the Description Logic SHIQ". In: *Journal of Artificial Intelligence Research* 31 (2008), pp. 157–204. DOI: 10.1613/jair.2372 (cit. on p. 15).

[76] Asunción Gómez-Pérez. "Ontology Evaluation". In: *Handbook on Ontologies*. Springer, 2004, pp. 251–273. DOI: 10.1007/978-3-540-24750-0_13 (cit. on pp. 49, 50).

[77] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer, 2006. DOI: 10.1007/b97353 (cit. on p. 20).

[78] Tudor Groza, Tania Tudorache, and Michel Dumontier. "State of the art and open challenges in community-driven knowledge curation". In: *Journal of Biomedical Informatics* 46.1 (2013), pp. 1–4. DOI: 10.1016/j.jbi.2012.11.007 (cit. on p. 51).

[79] Thomas R Gruber. "A translation approach to portable ontology specifications". In: *Knowledge acquisition* 5.2 (1993), pp. 199–220. DOI: 10.1006/knac.1993.1008 (cit. on p. 14).

[80] Nicola Guarino and Christopher Welty. "Evaluating Ontological Decisions with OntoClean". In: *Communications of the ACM* 45.2 (2002), pp. 61–65. DOI: 10.1145/503124.503150 (cit. on p. 49).

[81] Ramanathan V Guha, Dan Brickley, and Steve Macbeth. "Schema.org: evolution of structured data on the web". In: *Communications of the ACM* 59.2 (2016), pp. 44–51. DOI: 10.1145/2844544 (cit. on pp. 6, 17).

[82] Dominique Guinard and Vlad Trifa. *Building the Web of Things*. Manning Publications Co., 2016. ISBN: 1617292680 (cit. on p. 4).

[83] Jingzhi Guo, Li Da Xu, Guangyi Xiao, and Zhiguo Gong. “Improving Multilingual Semantic Interoperation in Cross-Organizational Enterprise Systems Through Concept Disambiguation”. In: *IEEE Transactions on Industrial Informatics* 8.3 (2012), pp. 647–658. DOI: 10.1109/TII.2012.2188899 (cit. on p. 52).

[84] Amelie Gyrard, Christian Bonnet, Karima Boudaoud, and Martin Serrano. “LOV4IoT: A second life for ontology-based domain knowledge to build Semantic Web of Things applications”. In: *IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE. 2016, pp. 254–261. DOI: 10.1109/FiCloud.2016.44 (cit. on pp. 8, 46, 67).

[85] Amelie Gyrard, Soumya Kanti Datta, and Christian Bonnet. “A survey and analysis of ontology-based software tools for semantic interoperability in IoT and WoT landscapes”. In: *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. IEEE. 2018, pp. 86–91. DOI: 10.1109/WF-IoT.2018.8355091 (cit. on p. 8).

[86] Amelie Gyrard, Soumya Kanti Datta, Christian Bonnet, and Karima Boudaoud. “Standardizing generic cross-domain applications in Internet of Things”. In: *2014 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2014, pp. 589–594. DOI: 10.1109/GLOCOMW.2014.7063496 (cit. on p. 8).

[87] Amelie Gyrard, Martin Serrano, and Ghislain A Atemezing. “Semantic Web Methodologies, Best Practices and Ontology Engineering Applied to Internet of Things”. In: *2nd World Forum on Internet of Things (WF-IoT)*. IEEE. 2015, pp. 412–417. DOI: 10.1109/WF-IoT.2015.7389090 (cit. on p. 49).

[88] Amelie Gyrard, Antoine Zimmermann, and Amit Sheth. “Building IoT-Based Applications for Smart Cities: How Can Ontology Catalogs Help?” In: *IEEE Internet of Things Journal* 5.5 (2018), pp. 3978–3990. DOI: 10.1109/JIOT.2018.2854278 (cit. on pp. 8, 72, 76).

[89] Sara Hachem, Thiago Teixeira, and Valérie Issarny. “Ontologies for the Internet of Things”. In: *Proceedings of the 8th Middleware Doctoral Symposium*. ACM. 2011, p. 3. DOI: 10.1145/2093190.2093193 (cit. on p. 66).

[90] Marina Harlamova, Mārīte Kirikova, and Kurt Sandkuhl. “A Survey on Challenges of Semantics Application in the Internet of Things Domain”. In: *Applied Computer Systems* 21.1 (2017), pp. 13–21. DOI: 10.1515/acss-2017-0002 (cit. on p. 9).

[91] Andreas Harth. “VisiNav: A System for Visual Search and Navigation on Web Data”. In: *Journal of Web Semantics* 8.4 (2010), pp. 348–354. DOI: 10.1016/j.websem.2010.08.001 (cit. on pp. 51, 52).

[92] Andreas Harth. “VisiNav: Visual Web Data Search and Navigation”. In: *Database and Expert Systems Applications*. Springer. 2009, pp. 214–228. DOI: 10.1007/978-3-642-03573-9_17 (cit. on pp. 35, 57).

[93] Andreas Harth, Aidan Hogan, Renaud Delbru, Jürgen Umbrich, Sean O'Riain, and Stefan Decker. "SWSE: Answers Before Links!" In: *Proceedings of the 2007 International Conference on Semantic Web Challenge*. Vol. 295. CEUR. 2007, pp. 137–144 (cit. on pp. 33, 38, 57).

[94] Andreas Harth, Jürgen Umbrich, and Stefan Decker. "MultiCrawler: A Pipelined Architecture for Crawling and Indexing Semantic Web Data". In: *International Semantic Web Conference*. Vol. 4273. Springer. 2006, pp. 258–271. DOI: 10.1007/11926078_19 (cit. on pp. 33, 49).

[95] Jens Hartmann, Raúl Palma, and Asunción Gómez-Pérez. "Ontology Repositories". In: *Handbook on Ontologies*. Springer, 2009, pp. 551–571. DOI: 10.1007/978-3-540-92673-3_25 (cit. on pp. 6, 30, 46, 84).

[96] Tom Heath and Christian Bizer. "Linked Data: Evolving the Web into a Global Data Space". In: *Synthesis Lectures on the Semantic Web: Theory and Technology* 1.1 (2011), pp. 1–136. DOI: 10.2200/S00334ED1V01Y201102WBE001 (cit. on pp. 6, 49, 50).

[97] Monika Henzinger. "Search Technologies for the Internet". In: *Science* 317.5837 (2007), pp. 468–471. DOI: 10.1126/science.1126557 (cit. on p. 52).

[98] Diane I Hillmann, Stuart A Sutton, Jon Phipps, and Ryan Laundry. "A Metadata Registry from Vocabularies Up: The NSDL Registry Project". In: *International Conference on Dublin Core and Metadata Applications* (2006) (cit. on pp. 32, 58).

[99] Hlomani Hlomani and Deborah Stacey. "Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey". In: (2014) (cit. on pp. 49, 50).

[100] Aidan Hogan. "The Semantic Web: Two Decades On". In: *The Semantic Web* (2020), p. 17 (cit. on p. 6).

[101] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, et al. "Knowledge Graphs". In: *arXiv preprint arXiv:2003.02320* (2020) (cit. on p. 15).

[102] Aidan Hogan, Stefan Decker, and Andreas Harth. "ReConRank: A Scalable Ranking Method for Semantic Web Data with Context". In: *Proceedings of Second International Workshop on Scalable Semantic Web Knowledge Base Systems*. 2006 (cit. on p. 38).

[103] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. "Weaving the Pedantic Web". In: *3rd International Workshop on Linked Data on the Web (LDOW2010)* (2010) (cit. on pp. 6, 49, 50).

[104] Aidan Hogan, Andreas Harth, Jürgen Umbrich, Sheila Kinsella, Axel Polleres, and Stefan Decker. "Searching and browsing Linked Data with SWSE: The Semantic Web Search Engine". In: *Journal of Web Semantics* 9.4 (2011), pp. 365–401. DOI: 10.1016/j.websem.2011.06.004 (cit. on p. 33).

[105] Kerrie Holley, Samuel Antoun, Ali Arsanjani, WA Brown, C Cozzi, JF Costas, et al. *The Power of the API Economy–Stimulate Innovation, Increase Productivity, Develop New Channels, and Reach New Markets*. Tech. rep. IBM Corporate, 2014 (cit. on p. 106).

[106] Ian Horrocks. "OWL: A Description Logic Based Ontology Language". In: *International conference on principles and practice of constraint programming*. Springer. 2005, pp. 5–8. DOI: 10.1007/11564751_2 (cit. on p. 14).

[107] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. "SWRL: A semantic web rule language combining OWL and RuleML". In: *W3C Member submission* 21 (2004), p. 79 (cit. on p. 16).

[108] Ian Horrocks and Ulrike Sattler. "Ontology Reasoning in the SHOQ(D) Description Logic". In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 1*. Vol. 1. 3. 2001, pp. 199–204 (cit. on p. 15).

[109] Krzysztof Janowicz, Armin Haller, Simon JD Cox, Danh Le Phuoc, and Maxime Lefrancois. "SOSA: A lightweight ontology for sensors, observations, samples, and actuators". In: *Journal of Web Semantics* (2018). DOI: 10.1016/j.websem.2018.06.003 (cit. on p. 17).

[110] Krzysztof Janowicz, Pascal Hitzler, Benjamin Adams, Dave Kolas, and Charles Vardeman II. "Five Stars of Linked Data Vocabulary Use". In: *Semantic Web* 5.3 (2014), pp. 173–176. DOI: 10.3233/SW-140135 (cit. on pp. 6, 49, 50).

[111] Kalervo Järvelin and Jaana Kekäläinen. "Cumulated Gain-Based Evaluation of IR Techniques". In: *ACM Transactions on Information Systems (TOIS)* 20.4 (2002), pp. 422–446. DOI: 10.1145/582415.582418 (cit. on pp. 26, 84, 99).

[112] Xing Jiang and Ah-Hwee Tan. "OntoSearch: A Full-Text Search Engine for the Semantic Web". In: *Proceedings of the 21st national conference on Artificial intelligence*. Vol. 2. AAAI. 2006, pp. 1325–1330 (cit. on p. 32).

[113] Matthew Jones and Harith Alani. "Content-based Ontology Ranking". In: *9th International Protégé Conference* (2006) (cit. on p. 53).

[114] Simon Jupp, Tony Burdett, Catherine Leroy, and Helen E Parkinson. "A New Ontology Lookup Service at EMBL-EBI". In: *Proceedings of Semantic Web Applications and Tools for Life Science International Conference*. CEUR, 2015, pp. 118–119 (cit. on pp. 37, 58).

[115] Megan Katsumi and Michael Grüninger. "Choosing ontologies for reuse". In: *Applied Ontology* 12.3-4 (2017), pp. 195–221. DOI: 10.3233/AO-160171 (cit. on pp. 21, 84).

[116] Megan Katsumi and Michael Grüninger. "What is ontology reuse?" In: *Formal Ontology in Information Systems*. IOS Press, 2016, pp. 9–22. DOI: 10.3233/978-1-61499-660-6-9 (cit. on p. 19).

[117] Michael Kifer. "Rule Interchange Format: The Framework". In: *International Conference on Web Reasoning and Rule Systems*. Springer. 2008, pp. 1–11. DOI: 10.1007/978-3-540-88737-9_1 (cit. on p. 16).

[118] Craig A Knoblock, Pedro Szekely, José Luis Ambite, Aman Goel, Shubham Gupta, Kristina Lerman, et al. "Semi-Automatically Mapping Structured Sources into the Semantic Web". In: *Extended Semantic Web Conference*. Springer. 2012, pp. 375–390. DOI: 10.1007/978-3-642-30284-8_32 (cit. on p. 66).

[119] Holger Knublauch and Dimitris Kontokostas. "Shapes constraint language (SHACL)". In: *W3C Recommendation* 11.8 (2017). URL: https://www.w3.org/TR/shacl/ (cit. on p. 16).

[120] Niklas Kolbe, Sylvain Kubler, and Yves Le Traon. "Popularity-driven Ontology Ranking using Qualitative Features". In: *International Semantic Web Conference*. Springer. 2019. DOI: 10.1007/978-3-030-30793-6_19 (cit. on pp. 10, 11, 71, 86, 87, 93, 102, 127).

[121] Niklas Kolbe, Sylvain Kubler, Jérémy Robert, Yves Le Traon, and Arkady Zaslavsky. "Linked Vocabulary Recommendation Tools for Internet of Things: A Survey". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), p. 127. DOI: 10.1145/3284316 (cit. on pp. 10, 11, 45, 74, 84, 87, 92, 94, 102, 127).

[122] Niklas Kolbe, Sylvain Kubler, Jérémy Robert, Yves Le Traon, and Arkady Zaslavsky. "Towards Semantic Interoperability in an Open IoT Ecosystem for Connected Vehicle Services". In: *Global Internet of Things Summit (GIoTS)*. IEEE. 2017. DOI: 10.1109/GIOTS.2017.8016229 (cit. on pp. 111, 128).

[123] Niklas Kolbe, Jérémy Robert, Sylvain Kubler, and Yves Le Traon. "PROFICIENT: Productivity Tool for Semantic Interoperability in an Open IoT Ecosystem". In: *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous)*. ACM. 2017. DOI: 10.1145/3144457.3144479 (cit. on pp. 11, 105, 127).

[124] Niklas Kolbe, Pierre-Yves Vandenbussche, Sylvain Kubler, and Yves Le Traon. "LOVBench: Ontology Ranking Benchmark". In: *Proceedings of The Web Conference (WWW)*. ACM, 2020. DOI: 10.1145/3366423.3380245 (cit. on pp. 11, 83, 127).

[125] Niklas Kolbe, Arkady Zaslavsky, Sylvain Kubler, Jérémy Robert, and Yves Le Traon. "Enriching a Situation Awareness Framework for IoT with Knowledge Base and Reasoning Components". In: *International and Interdisciplinary Conference on Modeling and Using Context*. Springer. 2017, pp. 41–54. DOI: 10.1007/978-3-319-57837-8_4 (cit. on p. 128).

[126] Maxim Kolchin, Nikolay Klimov, Alexey Andreev, Ivan Shilin, Daniil Garayzuev, Dmitry Mouromtsev, et al. "Ontologies for web of things: A pragmatic review". In: *International Conference on Knowledge Engineering and the Semantic Web*. Springer. 2015, pp. 102–116. DOI: 10.1007/978-3-319-24543-0_8 (cit. on pp. 72, 76).

[127] Sefki Kolozali, Maria Bermudez-Edo, Daniel Puschmann, Frieder Ganz, and Payam Barnaghi. "A Knowledge-based Approach for Real-Time IoT Data Stream Annotation and Processing". In: *IEEE International Conference on Internet of Things (iThings), Green Computing and Communications (GreenCom), and Cyber-Physical-Social Computing (CPSCom)*. IEEE. 2014, pp. 215–222. DOI: 10.1109/iThings.2014.39 (cit. on p. 66).

[128] Erno Kovacs, Martin Bauer, Jaeho Kim, Jaeseok Yun, Franck Le Gall, and Mengxuan Zhao. "Standards-Based Worldwide Semantic Interoperability for IoT". In: *IEEE Communications Magazine* 54.12 (2016), pp. 40–46. DOI: 10.1109/MCOM.2016.1600460CM (cit. on pp. 8, 66).

[129] Sylvain Kubler, Jérémy Robert, Ahmed Hefnawy, et al. "Open IoT Ecosystem for Sporting Event Management". In: *IEEE Access* 5.1 (2017), pp. 7064–7079. DOI: 10.1109/ACCESS.2017.2692247 (cit. on pp. 106, 108, 109).

[130] Yuangui Lei, Victoria Uren, and Enrico Motta. "Semsearch: A Search Engine for the Semantic Web". In: *International Conference on Knowledge Engineering and Knowledge Management*. Springer. 2006, pp. 238–245. DOI: 10.1007/11891451_22 (cit. on p. 41).

[131] Holger Lewen, Kaustubh Supekar, Natalya F Noy, and Mark A Musen. "Topic-Specific Trust and Open Rating Systems: An Approach for Ontology Evaluation". In: *4th International Workshop on Evaluation of Ontologies for the Web*. CEUR, 2006 (cit. on pp. 32, 57).

[132] Tie-Yan Liu. "Learning to Rank for Information Retrieval". In: *Foundations and Trends in Information Retrieval* 3.3 (2009), pp. 225–331. DOI: 10.1007/978-3-642-14267-3 (cit. on pp. 23, 24, 54, 63, 73, 77, 84, 88, 97, 98).

[133] Adolfo Lozano-Tello and Asunción Gómez-Pérez. "ONTOMETRIC: A Method to Choose the Appropriate Ontology". In: *Journal of Database Management (JDM)* 2.15 (2004), pp. 1–18. DOI: 10.4018/jdm.2004040101 (cit. on pp. 31, 57).

[134] Craig Macdonald, Rodrygo LT Santos, and Iadh Ounis. "The whens and hows of learning to rank for web search". In: *Information Retrieval* 16.5 (2013), pp. 584–628. DOI: 10.1007/s10791-012-9209-9 (cit. on pp. 24, 97).

[135] Alexander Maedche and Steffen Staab. "Ontology learning for the Semantic Web". In: *IEEE Intelligent Systems* 16.2 (2001), pp. 72–79. DOI: 10.1109/5254.920602 (cit. on p. 20).

[136] Alexander Maedche and Valentin Zacharias. "Clustering Ontology-Based Metadata in the Semantic Web". In: *PKDD*. Vol. 2. Springer. 2002, pp. 348–360. DOI: 10.1007/3-540-45681-3_29 (cit. on p. 15).

[137] Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, Peyman Adibi, Payam Barnaghi, and Amit P Sheth. "Machine learning for Internet of Things data analysis: A survey". In: *Digital Communications and Networks* 4.3 (2018), pp. 161–175. DOI: 10.1016/j.dcan.2017.10.002 (cit. on p. 17).

[138] Bogdan Manate, Victor Ion Munteanu, and Teodor Florin Fortis. "Towards a Smarter Internet of Things: Semantic Visions". In: *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*. IEEE. 2014, pp. 582–587. DOI: 10.1109/CISIS.2014.84 (cit. on p. 9).

[139] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *An Introduction to Information Retrieval*. Cambridge University Press, 2008. ISBN: 0521865719 (cit. on pp. 22, 24–26, 53, 54).

[140] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, et al. "OWL-S: Semantic markup for web services". In: *W3C member submission* 22.4 (2004) (cit. on p. 16).

[141] Marcos Martínez-Romero, Clement Jonquet, Martin J O'Connor, John Graybeal, Alejandro Pazos, and Mark A Musen. "NCBO Ontology Recommender 2.0: an enhanced approach for biomedical ontology recommendation". In: *Journal of Biomedical Semantics* 8.1 (2017), p. 21. DOI: `10.1186/s13326-017-0128-y` (cit. on pp. 40, 41, 59, 79, 82, 93, 122).

[142] Marcos Martínez-Romero, José M Vázquez-Naya, Javier Pereira, and Alejandro Pazos. "BiOSS: A system for biomedical ontology selection". In: *Computer Methods and Programs in Biomedicine* 114.1 (2014), pp. 125–140. DOI: `10.1016/j.cmpb.2014.01.020` (cit. on pp. 36, 58, 102).

[143] Nicolas Matentzoglu, Daniel Tang, Bijan Parsia, and Uli Sattler. "The Manchester OWL Repository: System Description". In: *Proceedings of the 2014 International Semantic Web Conference on Posters & Demonstrations Track*. CEUR. 2014, pp. 285–288 (cit. on pp. 37, 58).

[144] Brian McBride. "The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS". In: *Handbook on Ontologies*. Springer, 2004, pp. 51–65. DOI: `10.1007/978-3-540-24750-0_3` (cit. on p. 14).

[145] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in action: covers Apache Lucene 3.0*. Manning Publications Co., 2010. ISBN: 1933988177 (cit. on pp. 74, 84).

[146] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*. 2013, pp. 3111–3119 (cit. on p. 74).

[147] Eric Miller. "An Introduction to the Resource Description Framework". In: *Bulletin of the Association for Information Science and Technology* 25.1 (1998), pp. 15–19. DOI: `10.1002/bult.105` (cit. on pp. 6, 14).

[148] George A Miller. "WordNet: a lexical database for English". In: *Communications of the ACM* 38.11 (1995), pp. 39–41. DOI: `10.1145/219717.219748` (cit. on pp. 30, 52, 75).

[149] Julien Mineraud, Oleksiy Mazhelis, Xiang Su, and Sasu Tarkoma. "A gap analysis of Internet-of-Things platforms". In: *Computer Communications* 89 (2016), pp. 5–16. DOI: `10.1016/j.comcom.2016.03.015` (cit. on pp. 9, 65, 66).

[150] Tom Minka and Stephen Robertson. "Selection bias in the LETOR datasets". In: *SIGIR Workshop on Learning to Rank for Information Retrieval*. 2008, pp. 48–51 (cit. on pp. 26, 97).

[151] Michael Mrissa, Lionel Médini, and Jean-Paul Jamont. "Semantic Discovery and Invocation of Functionalities for the Web of Things". In: *23rd International WETICE Conference (WETICE)*. IEEE. 2014, pp. 281–286. DOI: `10.1109/WETICE.2014.50` (cit. on p. 66).

[152] Paul Murdock, Louay Bassbouss, Martin Bauer, Mahdi Ben Alaya, Rajdeep Bhowmik, Patricia Brett, et al. *Semantic interoperability for the Web of Things*. Tech. rep. 2016 (cit. on p. 4).

[153] Sebastian Neumaier, Axel Polleres, Simon Steyskal, and Jürgen Umbrich. "Data Integration for Open Data on the Web". In: *Reasoning Web International Summer School*. Springer. 2017, pp. 1–28. DOI: 10.1007/978-3-319-61033-7_1 (cit. on p. 63).

[154] Anne H Ngu, Mario Gutierrez, Vangelis Metsis, Surya Nepal, and Quan Z Sheng. "IoT Middleware: A Survey on Issues and Enabling Technologies". In: *IEEE Internet of Things Journal* 4.1 (2016), pp. 1–20. DOI: 10.1109/JIOT.2016.2615180 (cit. on p. 8).

[155] Mahda Noura, Mohammed Atiquzzaman, and Martin Gaedke. "Interoperability in internet of things: Taxonomies and open challenges". In: *Mobile Networks and Applications* 24.3 (2019), pp. 796–809. DOI: 10.1007/s11036-018-1089-9 (cit. on p. 9).

[156] Natalya F Noy, Paul R Alexander, Rave Harpaz, Patricia L Whetzel, Raymond W Fergerson, and Mark A Musen. "Getting Lucky in Ontology Search: A Data-Driven Evaluation Framework for Ontology Ranking". In: *International Semantic Web Conference*. Springer. 2013, pp. 444–459. DOI: 10.1007/978-3-642-41335-3_28 (cit. on pp. 41, 122).

[157] Natalya F Noy, Nigam H Shah, Patricia L Whetzel, Benjamin Dai, Michael Dorf, Nicholas Griffith, et al. "BioPortal: ontologies and integrated data resources at the click of a mouse". In: *Nucleic Acids Research* 37 (2009). DOI: 10.1093/nar/gkp440 (cit. on pp. 6, 35, 58).

[158] Natasha Noy, Matthew Burgess, and Dan Brickley. "Google Dataset Search: Building a search engine for datasets in an open Web ecosystem". In: *The World Wide Web Conference*. ACM. 2019, pp. 1365–1375. DOI: 10.1145/3308558.3313685 (cit. on p. 6).

[159] Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. "Industry-Scale Knowledge Graphs: Lessons and Challenges". In: *Communications of the ACM* 62.8 (2019), pp. 36–43. DOI: 10.1145/3331166 (cit. on p. 6).

[160] Marcelo Iury S Oliveira and Bernadette Farias Lóscio. "What is a data ecosystem?" In: *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*. 2018, pp. 1–9. DOI: 10.1145/3209281.3209335 (cit. on p. 4).

[161] Edison Ong, Zuoshuang Xiang, Bin Zhao, Yue Liu, Yu Lin, Jie Zheng, et al. "Ontobee: A linked ontology data server to support ontology term dereferencing, linkage, query and integration". In: *Nucleic Acids Research* 45.D1 (2017), pp. D347–D352. DOI: 10.1093/nar/gkw918 (cit. on p. 36).

[162] Eyal Oren, Renaud Delbru, Michele Catasta, Richard Cyganiak, Holger Stenzhorn, and Giovanni Tummarello. "Sindice.com: a document-oriented lookup index for open linked data". In: *International Journal of Metadata, Semantics and Ontologies* 3.1 (2008), pp. 37–52. DOI: 10.1504/IJMSO.2008.021204 (cit. on p. 33).

[163] Lorena Otero-Cerdeira, Francisco J Rodríguez-Martínez, and Alma Gómez-Rodríguez. "Ontology matching: A literature review". In: *Expert Systems with Applications* 42.2 (2015), pp. 949–971. DOI: 10.1016/j.eswa.2014.08.032 (cit. on p. 8).

[164] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Stanford InfoLab, 1999 (cit. on pp. 23, 53, 75).

[165] Raúl Palma, Peter Haase, and Asunción Gómez-Pérez. "Oyster: Sharing and Re-using Ontologies in a Peer-to-Peer Community". In: *Proceedings of the 15th International Conference on World Wide Web*. ACM. 2006, pp. 1009–1010. DOI: 10.1145/1135777.1135989 (cit. on pp. 33, 58).

[166] Jeff Z Pan, Edward Thomas, and Derek Sleeman. "ONTOSEARCH2: Searching and Querying Web Ontologies". In: *Proceedings of the IADIS International Conference on WWW/Internet* 2006 (2006), pp. 211–218 (cit. on pp. 32, 58).

[167] Chintan Patel, Kaustubh Supekar, Yugyung Lee, and EK Park. "OntoKhoj: A Semantic Web Portal for Ontology Searching, Ranking and Classification". In: *Proceedings of the 5th ACM International Workshop on Web Information and Data Management*. ACM. 2003, pp. 58–61. DOI: 10.1145/956699.956712 (cit. on pp. 30, 57).

[168] Pankesh Patel, Amelie Gyrard, Soumya Kanti Datta, and Muhammad Intizar Ali. "SWoTSuite: A Toolkit for Prototyping End-to-End Semantic Web of Things Applications". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. 2017, pp. 263–267. DOI: 10.1145/3041021.3054736 (cit. on p. 66).

[169] Charith Perera. "Sensing as a Service (S2aaS): Buying and Selling IoT Data". In: *IEEE IoT Newsletter* (2016) (cit. on pp. 4, 5).

[170] Charith Perera, Chi Harold Liu, and Srimal Jayawardena. "The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey". In: *IEEE Transactions on Emerging Topics in Computing* 3.4 (2015), pp. 585–598. DOI: 10.1109/TETC.2015.2390034 (cit. on p. 4).

[171] Charith Perera, Yongrui Qin, Julio C Estrella, Stephan Reiff-Marganiec, and Athanasios V Vasilakos. "Fog Computing for Sustainable Smart Cities: A Survey". In: *ACM Computing Surveys (CSUR)* 50.3 (2017), p. 32. DOI: 10.1145/3057266 (cit. on p. 9).

[172] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. "Context Aware Computing for The Internet of Things: A Survey". In: *IEEE Communications Surveys & Tutorials* 16.1 (2014), pp. 414–454. DOI: 10.1109/SURV.2013.042313.00197 (cit. on pp. 17, 22).

[173] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. "Sensing as a Service Model for Smart Cities Supported by Internet of Things". In: *Transactions on Emerging Telecommunications Technologies* 25.1 (2014), pp. 81–93. DOI: 10.1002/ett.2704 (cit. on p. 6).

[174] Dennis Pfisterer, Kay Romer, Daniel Bimschas, Oliver Kleine, Richard Mietz, Cuong Truong, et al. "SPITFIRE: Toward a Semantic Web of Things". In: *IEEE Communications Magazine* 49.11 (2011), pp. 40–48. DOI: 10.1109/MCOM.2011.6069708 (cit. on pp. 8, 22, 66).

[175] María Poveda Villalón, Raul García Castro, and A Gómez-Pérez. "Building an Ontology Catalogue for Smart Cities". In: CRC Press, 2014, pp. 829–839 (cit. on pp. 8, 37, 58).

[176] Maria Poveda Villalon, Mari Carmen Suárez-Figueroa, and Asunción Gómez-Pérez. "A Double Classification of Common Pitfalls in Ontologies". In: *Workshop on Ontology Quality (OntoQual 2010)* (2010) (cit. on p. 49).

[177] Eric Prud, Andy Seaborne, et al. "SPARQL Query Language for RDF". In: (2006) (cit. on p. 16).

[178] Eric Prudhommeaux and Ryan Lee. *W3C RDF validation service*. 2007. URL: https://www.w3.org/RDF/Validator/ (cit. on p. 50).

[179] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. "LETOR: A benchmark collection for research on learning to rank for information retrieval". In: *Information Retrieval* 13.4 (2010), pp. 346–374. DOI: 10.1007/s10791-009-9123-y (cit. on pp. 87, 97).

[180] Yongrui Qin, Quan Z Sheng, Nickolas JG Falkner, Schahram Dustdar, Hua Wang, and Athanasios V Vasilakos. "When things matter: A survey on data-centric internet of things". In: *Journal of Network and Computer Applications* 64 (2016), pp. 137–153. DOI: 10.1016/j.jnca.2015.12.016 (cit. on p. 9).

[181] Yuzhong Qu and Gong Cheng. "Falcons Concept Search: A Practical Search Engine for Web Ontologies". In: *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 41.4 (2011), pp. 810–816. DOI: 10.1109/TSMCA.2011.2132705 (cit. on pp. 34, 41, 57).

[182] Filip Radulovic, María Poveda-Villalón, Daniel Vila-Suero, Víctor Rodríguez-Doncel, Raúl García-Castro, and Asunción Gómez-Pérez. "Guidelines for Linked Data generation and publication: An example in building energy consumption". In: *Automation in Construction* 57 (2015), pp. 178–187. DOI: 10.1016/j.autcon.2015.04.002 (cit. on pp. 6, 49).

[183] D Raggett. "The Web of Things: Challenges and Opportunities". In: *Computer* 5 (2015), pp. 26–32. DOI: 10.1109/MC.2015.149 (cit. on p. 4).

[184] Joshi Rajive, Paul Didier, Jaime Jimenez, et al. *The Industrial Internet of Things Volume G5: Connectivity Framework*. Tech. rep. Industrial Internet Consortium, 2017 (cit. on pp. 5, 6, 106).

[185] Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, and Siobhán Clarke. "Middleware for Internet of Things: A Survey". In: *IEEE Internet of things journal* 3.1 (2015), pp. 70–95. DOI: 10.1109/JIOT.2015.2498900 (cit. on p. 8).

[186] Antonio J Roa-Valverde and Miguel-Angel Sicilia. "A survey of approaches for ranking on the web of data". In: *Information Retrieval* 17.4 (2014), pp. 295–325. DOI: 10.1007/s10791-014-9240-0 (cit. on pp. 52, 84).

[187] Jérémy Robert, Sylvain Kubler, Niklas Kolbe, Alessandro Cerioni, Emmanuel Gastaud, and Kary Främling. "Open IoT Ecosystem for Enhanced Interoperability in Smart Cities–Example of Métropole De Lyon". In: *Sensors* 17.12 (2017), p. 2849. DOI: 10.3390/s17122849 (cit. on pp. 4, 127).

[188] Stephen E Robertson. "Overview of the okapi projects". In: *Journal of documentation* 53.1 (1997), pp. 3–7. DOI: 10.1108/EUM0000000007186 (cit. on pp. 23, 93).

[189] Cristiano Rocha, Daniel Schwabe, and Marcus Poggi Aragao. "A Hybrid Approach for Searching in the Semantic Web". In: *Proceedings of the 13th International Conference on World Wide Web*. ACM. 2004, pp. 374–383. DOI: 10.1145/988672.988723 (cit. on p. 41).

[190] Kurt Rohloff, Mike Dean, Ian Emmons, Dorene Ryder, and John Sumner. "An Evaluation of Triple-Store Technologies for Large Data Stores". In: *On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops*. Springer. 2007, pp. 1105–1114. DOI: 10.1007/978-3-540-76890-6_38 (cit. on p. 16).

[191] Carlos Rueda, Luis Bermudez, and Janet Fredericks. "The MMI Ontology Registry and Repository: A Portal for Marine Metadata Interoperability". In: *Biloxi-Marine Technology for Our Future: Global and Local Challenges*. IEEE. 2009, pp. 1–6. DOI: 10.23919/OCEANS.2009.5422206 (cit. on pp. 35, 58).

[192] Thomas L Saaty. "Decision making with the analytic hierarchy process". In: *International Journal of Services Sciences* 1.1 (2008), pp. 83–98. DOI: 10.1504/IJSSCI.2008.017590 (cit. on p. 54).

[193] Marta Sabou, Vanessa Lopez, Enrico Motta, and Victoria Uren. "Ontology Selection: Ontology Evaluation on the Real Semantic Web". In: *4th International Workshop on Evaluation of Ontologies for the Web*. 2006 (cit. on pp. 30, 46, 49–52, 72).

[194] Gerard Salton and Christopher Buckley. "Term-weighting approaches in automatic text retrieval". In: *Information processing & management* 24.5 (1988), pp. 513–523. DOI: 10.1016/0306-4573(88)90021-0 (cit. on pp. 23, 53).

[195] Gerard Salton, Anita Wong, and Chung-Shu Yang. "A Vector Space Model for Automatic Indexing". In: *Communications of the ACM* 18.11 (1975), pp. 613–620. DOI: 10.1145/361219.361220 (cit. on pp. 54, 97).

[196] Johann Schaible, Thomas Gottron, and Ansgar Scherp. "Survey on Common Strategies of Vocabulary Reuse in Linked Open Data Modeling". In: *European Semantic Web Conference*. Springer. 2014, pp. 457–472. DOI: 10.1007/978-3-319-07443-6_31 (cit. on pp. 27, 53, 63, 72, 93, 122).

[197] Johann Schaible, Thomas Gottron, and Ansgar Scherp. "TermPicker: Enabling the Reuse of Vocabulary Terms by Exploiting Data from the Linked Open Data Cloud". In: *International Semantic Web Conference*. Springer. 2016, pp. 101–117. DOI: 10.1007/978-3-319-34129-3_7 (cit. on pp. 39, 42, 58, 66, 72, 86, 93).

[198] François Scharffe, Ghislain Atemezing, Raphaël Troncy, Fabien Gandon, Serena Villata, Bénédicte Bucher, et al. "Enabling Linked Data Publication with the Datalift Platform". In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence* (cit. on p. 46).

[199] Souvik Sengupta, Jordi Garcia, and Xavi Masip-Bruin. "A Literature Survey on Ontology of Different Computing Platforms in Smart Environments". In: (2018) (cit. on p. 8).

[200] Omer Berat Sezer, Erdogan Dogdu, and Ahmet Murat Ozbayoglu. "Context-Aware Computing, Learning, and Big Data in Internet of Things: A Survey". In: *IEEE Internet of Things Journal* 5.1 (2017), pp. 1–27. DOI: 10.1109/JIOT.2017.2773600 (cit. on pp. 17, 22).

[201] Rob Shearer, Boris Motik, and Ian Horrocks. "HermiT: A Highly-Efficient OWL Reasoner". In: *Proceedings of the Fifth OWLED Workshop on OWL: Experiences and Directions*. Vol. 432. CEUR. 2008, p. 91 (cit. on p. 16).

[202] Quan Z Sheng, Xiaoqiang Qiao, Athanasios V Vasilakos, et al. "Web services composition: A decades overview". In: *Information Sciences* 280 (2014), pp. 218–238. DOI: 10.1016/j.ins.2014.04.054 (cit. on p. 108).

[203] Feifei Shi, Qingjuan Li, Tao Zhu, and Huansheng Ning. "A Survey of Data Semantization in Internet of Things". In: *Sensors* 18.1 (2018), p. 313. DOI: 10.3390/s18010313 (cit. on pp. 7, 9, 46).

[204] Elena Simperl. "Reusing ontologies on the Semantic Web: A feasibility study". In: *Data & Knowledge Engineering* 68.10 (2009), pp. 905–925. DOI: 10.1016/j.datak.2009.02.002 (cit. on pp. 6, 8, 19, 46, 72).

[205] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. "Pellet: A practical OWL-DL reasoner". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 5.2 (2007), pp. 51–53. DOI: 10.1016/j.websem.2007.03.004 (cit. on p. 15).

[206] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, et al. "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration". In: *Nature Biotechnology* 25.11 (2007), p. 1251. DOI: 10.1038/nbt1346 (cit. on pp. 34, 58, 84).

[207] Pablo Sotres, Jorge Lanza, Luis Sánchez, Juan Ramón Santana, Carmen López, and Luis Muñoz. "Breaking Vendors and City Locks through a Semantic-enabled Global Interoperable Internet-of-Things System: A Smart Parking Case". In: *Sensors* 19.2 (2019), p. 229. DOI: 10.3390/s19020229 (cit. on p. 4).

[208] Steffen Stadtmüller, Andreas Harth, and Marko Grobelnik. "Accessing information about Linked Data vocabularies with vocab.cc". In: *Semantic Web and Web Science*. Springer, 2013, pp. 391–396. DOI: 10.1007/978-1-4614-6880-6_34 (cit. on pp. 36, 57, 72).

[209] John A Stankovic. "Research Directions for the Internet of Things". In: *IEEE Internet of Things Journal* 1.1 (2014), pp. 3–9. DOI: 10.1109/JIOT.2014.2312291 (cit. on p. 4).

[210] Ioannis Stavrakantonakis, Anna Fensel, and Dieter Fensel. "Linked Open Vocabulary ranking and terms discovery". In: *Proceedings of the 12th International Conference on Semantic Systems*. ACM. 2016, pp. 1–8. DOI: 10.1145/2993318.2993338 (cit. on p. 81).

[211] Ioannis Stavrakantonakis, Anna Fensel, and Dieter Fensel. "Linked Open Vocabulary Recommendation based on Ranking and Linked Open Data". In: *Joint International Semantic Technology Conference*. Springer. 2015, pp. 40–55. DOI: 10.1007/978-3-319-31676-5_3 (cit. on pp. 39, 59).

[212] Darijus Strasunskas and Stein L Tomassen. "On Significance of Ontology Quality in Ontology-driven Web Search". In: *Emerging Technologies and Information Systems for the Knowledge Society* (cit. on pp. 49, 50).

[213] Kehua Su, Jie Li, and Hongbo Fu. "Smart city and the applications". In: *2011 International Conference on Electronics, Communications and Control (ICECC)*. IEEE. 2011, pp. 1028–1031 (cit. on p. 4).

[214] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, Enrico Motta, and Aldo Gangemi. *Ontology Engineering in a Networked World*. Springer, 2012. DOI: 10.1007/978-3-642-24794-1_1 (cit. on p. 20).

[215] R Subhashini, J Akilandeswari, and V Sinthuja. "A Review on Ontology Ranking Algorithms". In: *International Journal of Computer Applications* 33.4 (2011), pp. 6–11. DOI: 10.5120/4006-5677 (cit. on p. 54).

[216] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. *Vision and Challenges for Realising the Internet of Things*. Vol. 3. 3. Publications Office of the European Union, 2010, pp. 34–36. DOI: 10.2759/26127 (cit. on pp. 4, 8).

[217] Kaustubh Supekar, Chintan Patel, and Yugyung Lee. "Characterizing Quality of Knowledge on Semantic Web". In: *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference*. AAAI. 2004, pp. 472–478 (cit. on pp. 31, 50, 51, 57).

[218] Kaustubh Supekar, Daniel Rubin, Natasha Noy, and Mark Musen. "Knowledge Zone: A Public Repository of Peer-Reviewed Biomedical Ontologies". In: *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*. IOS Press. 2007, pp. 812–816. ISBN: 9781586037741 (cit. on pp. 32, 57).

[219] Ioan Szilagyi and Patrice Wira. "Ontologies and Semantic Web for the Internet of Things – a survey". In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2016, pp. 6949–6954. DOI: 10.1109/IECON.2016.7793744 (cit. on pp. 7, 46).

[220] Ruben Taelman, Miel Vander Sande, and Ruben Verborgh. "GraphQL-LD: Linked Data Querying with GraphQL". In: *Proceedings of the 17th International Semantic Web Conference: Posters and Demos*. 2018, pp. 1–4 (cit. on p. 16).

[221] Wei Tan, Yushun Fan, Ahmed Ghoneim, et al. "From the Service-Oriented Architecture to the Web API Economy". In: *IEEE Internet Computing* 20.4 (2016), pp. 64–68. DOI: 10.1109/MIC.2016.74 (cit. on p. 106).

[222] Samir Tartir and I Budak Arpinar. "Ontology Evaluation and Ranking using OntoQA". In: *International Conference on Semantic Computing*. IEEE. 2007, pp. 185–192. DOI: 10.1109/ICSC.2007.19 (cit. on p. 31).

[223] Samir Tartir, I Budak Arpinar, Michael Moore, Amit P Sheth, and Boanerges Aleman-Meza. "OntoQA: Metric-based Ontology Quality Analysis". In: 2005 (cit. on pp. 31, 57).

[224] Samir Tartir, I Budak Arpinar, and Amit P Sheth. "Ontological Evaluation and Validation". In: *Theory and Applications of Ontology: Computer Applications*. Springer, 2010, pp. 115–130. DOI: 10.1007/978-90-481-8847-5_5 (cit. on p. 49).

[225] Edward Thomas, Jeff Z Pan, and Derek H Sleeman. "ONTOSEARCH2: Searching Ontologies Semantically". In: *Proceedings of the OWLED Workshop on OWL: Experiences and Directions*. 2007 (cit. on p. 32).

[226] Andreas Tolk, Saikou Diallo, and Charles Turnitsa. "Applying the Levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and Composability for System-of-Systems Engineering". In: *Journal of Systemics, Cybernetics and Informatics* 8.5 (2007) (cit. on p. 5).

[227] Andreas Tolk and James A Muguira. "The Levels of Conceptual Interoperability Model". In: *Proceedings of the 2003 fall simulation interoperability workshop*. Vol. 7. 2003, pp. 1–11 (cit. on p. 5).

[228] Nguyen Khoi Tran, Quan Z Sheng, Muhammad Ali Babar, and Lina Yao. "Searching the Web of Things: State of the Art, Challenges, and Solutions". In: *ACM Computing Surveys (CSUR)* 50.4 (2017), p. 55. DOI: 10.1145/3092695 (cit. on p. 9).

[229] Giovanni Tummarello, Renaud Delbru, and Eyal Oren. "Sindice.com: Weaving the Open Linked Data". In: *The Semantic Web. ISWC 2007*. Springer, 2007, pp. 552–565. DOI: 10.1007/978-3-540-76298-0_40 (cit. on pp. 33, 57).

[230] Rachanee Ungrangsi, Chutiporn Anutariya, and Vilas Wuwongse. "combiSQORE: An Ontology Combination Algorithm". In: *The Semantic Web* (2007), pp. 566–579. DOI: 10.1007/978-3-540-76298-0_41 (cit. on pp. 34, 59).

[231] Rachanee Ungrangsi, Chutiporn Anutariya, and Vilas Wuwongse. "SQORE: an ontology retrieval framework for the next generation Web". In: *Concurrency and Computation: Practice and Experience* 21.5 (2009), pp. 651–671. DOI: 10.1002/cpe.1385 (cit. on pp. 34, 59).

[232] Pierre-Yves Vandenbussche, Ghislain A Atemezing, María Poveda-Villalón, and Bernard Vatant. "Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web". In: *Semantic Web* 8.3 (2017), pp. 437–452. DOI: 10.3233/SW-160213 (cit. on pp. 10, 38, 58, 63, 67, 72, 74, 79, 82, 84, 94, 96, 100, 101).

[233] Ovidiu Vermesan and Peter Friess. *Building the Hyperconnected Society: Internet of Things Research and Innovation Value Chains, Ecosystems and Markets*. Vol. 43. River Publishers, 2015. DOI: 10.13052/rp-9788793237988 (cit. on p. 4).

[234] Ovidiu Vermesan and Peter Friess. *Digitising the Industry: Internet of Things Connecting the Physical, Digital and Virtual Worlds*. River Publishers, 2016. DOI: 10.13052/rp-9788793379824 (cit. on pp. 6, 106).

[235] Kim Viljanen, Jouni Tuominen, Eetu Mäkelä, and Eero Hyvönen. "Normalized Access to Ontology Repositories". In: *IEEE Sixth International Conference on Semantic Computing (ICSC)*. IEEE. 2012, pp. 109–116. DOI: 10.1109/ICSC.2012.56 (cit. on pp. 30, 46).

[236] Chao Wang, Yiqun Liu, and Shaoping Ma. "Building a click model: From idea to practice". In: *CAAI Transactions on Intelligence Technology* 1.4 (2016), pp. 313–322. DOI: 10.1016/j.trit.2016.12.003 (cit. on pp. 24, 87).

[237] Xiang Wang, Xiaoming Zhang, and Mei Li. "A Survey on Semantic Sensor Web: Sensor Ontology, Mapping and Query". In: *International Journal of u- and e-Service, Science and Technology* 8.10 (2015), pp. 325–342. DOI: 0.14257/ijunesst.2015.8.10.32 (cit. on p. 8).

[238] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific data* 3 (2016). DOI: 10.1038/sdata.2016.18 (cit. on p. 7).

[239] Gang Wu, Juanzi Li, Ling Feng, and Kehong Wang. "Identifying Potentially Important Concepts and Relations in an Ontology". In: *International Semantic Web Conference*. Springer. 2008, pp. 33–49. DOI: 10.1007/978-3-540-88564-1_3 (cit. on pp. 41, 86, 92).

[240] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. "Adapting boosting for information retrieval measures". In: *Information Retrieval* 13.3 (2010), pp. 254–270. DOI: 10.1007/s10791-009-9112-1 (cit. on p. 54).

[241] Zuoshuang Xiang, Chris Mungall, Alan Ruttenberg, and Yongqun He. "Ontobee: A Linked Data Server and Browser for Ontology Terms". In: *International Conference on Biomedical Ontology*. 2011. DOI: 10.1093/nar/gkw918 (cit. on pp. 36, 58).

[242] Guangyi Xiao, Jingzhi Guo, Zhiguo Gong, and Renfa Li. "Semantic input method of Chinese word senses for semantic document exchange in e-business". In: *Journal of Industrial Information Integration* 3 (2016), pp. 31–36. DOI: 10.1016/j.jii.2016.07.002 (cit. on p. 52).

[243] Jun Xu and Hang Li. "Adarank: A Boosting Algorithm for Information Retrieval". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 2007, pp. 391–398. DOI: 10.1145/1277741.1277809 (cit. on p. 99).

[244] Juan Ye, Stamatia Dasiopoulou, Graeme Stevenson, Georgios Meditskos, Efstratios Kontopoulos, Ioannis Kompatsiaris, et al. "Semantic web technologies in pervasive computing: A survey and research roadmap". In: *Pervasive and Mobile Computing* 23 (2015), pp. 1–25. DOI: 10.1016/j.pmcj.2014.12.009 (cit. on pp. 9, 68).

[245] Emine Yilmaz and Stephen Robertson. "Deep versus shallow judgments in learning to rank". In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM. 2009, pp. 662–663. DOI: 10.1145/1571941.1572066 (cit. on p. 26).

[246] Andrea Zanella, Nicola Bui, Angelo Castellani, et al. "Internet of Things for Smart Cities". In: *IEEE Internet of Things journal* 1.1 (2014), pp. 22–32. DOI: 10.1109/JIOT.2014.2306328 (cit. on p. 4).

[247] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. "Quality Assessment for Linked Data: A Survey". In: *Semantic Web* 7.1 (2016), pp. 63–93. DOI: 10.3233/SW-150175 (cit. on pp. 49, 50, 74, 87, 92).

[248] Yi Zhang, Wamberto Vasconcelos, and Derek Sleeman. "OntoSearch: An Ontology Search Engine". In: *Research and Development in Intelligent Systems XXI*. Springer, 2005, pp. 58–69. DOI: 10.1007/1-84628-102-4_5 (cit. on pp. 32, 57).

[249] Anneke Zuiderwijk, Marijn Janssen, and Chris Davis. "Innovation with open data: Essential elements of open data ecosystems". In: *Information Polity* 19.1, 2 (2014), pp. 17–33. DOI: 10.3233/IP-140329 (cit. on p. 4).