Strategies and Techniques for Federated Semantic Knowledge Retrieval and Integration

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Diego Collarana Vargas
aus
Oruro, Bolivien

Bonn, 10.12.2018

Dieser Forschungsbericht wurde als Dissertation von der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Bonn angenommen und ist auf dem Hochschulschriftenserver der ULB Bonn http://hss.ulb.uni-bonn.de/diss_online elektronisch publiziert.

1. Gutachter:	Prof. Dr. Sören Auer
2. Gutachter:	Prof. Dr. Jens Lehmann

Tag der Promotion:	10.12.2018
Erscheinungsjahr:	2019

Abstract

The vast amount of data shared on the Web requires effective and efficient techniques to retrieve and create machine usable knowledge out of it. The creation of integrated knowledge from the Web, especially knowledge about the same entity spread over different web data sources, is a challenging task. Several data interoperability problems such as schema, structure, or domain conflicts need to be solved during the integration process. Semantic Web Technologies have evolved as a novel approach to tackle the problem of knowledge integration out of heterogeneous data. However, knowledge retrieval and integration from web data sources is an expensive process, mainly due to the Extraction-Transformation-Load approach that predominates the process. In addition, there are increasingly many scenarios, where a full physical integration of the data is either prohibitive (e.g. due to data being hidden behind APIs) or not allowed (e.g. for data privacy concerns). Thus, a more cost-effective and federated integration approach is needed, a method that supports organizations to create valuable insights out of the heterogeneous data spread on web sources. In this thesis, we tackle the problem of knowledge retrieval an integration from heterogeneous web sources and propose a holistic semantic knowledge retrieval and integration approach that creates knowledge graphs on-demand from a federation of web sources. We focus on the representation of web sources data, which belongs to the same entity, as pieces of knowledge to then synthesize them as knowledge graph solving interoperability conflicts at integration time. First, we propose MINTE, a novel semantic integration approach that solves interoperability conflicts present in heterogeneous web sources. MINTE defines the concept of RDF molecules to represent web sources data as pieces of knowledge. Then, MINTE relies on a semantic similarity function to determine RDF molecules belonging to the same entity. Finally, MINTE employs fusion policies for the synthesis of RDF molecules into a knowledge graph. Second, we define a similarity framework for RDF molecules to identify semantically equivalent entities. The framework includes state-of-the-art semantic similarity metrics, such as GADES, but also a semantic similarity metric based on embeddings named MateTee developed in the scope of this thesis. Ultimately, based on MINTE and our similarity framework, we design a federated semantic retrieval engine named FuhSen. FuhSen is able to effectively integrate data from heterogeneous web data sources and create an integrated knowledge graphs on-demand. FuhSen is equipped with a faceted browsing user interface oriented to facilitate the exploration of on-demand built knowledge graphs. We conducted several empirical evaluations to assess the effectiveness and efficiency of our holistic approach. More importantly, three domain applications, i.e., Law Enforcement, Job Market Analysis, and Manufacturing, have been developed and managed by our approach. Both the empirical evaluations and concrete applications provide evidence that the methodology and techniques proposed in this thesis help to effectively integrate the pieces of knowledge about entities that are spread over heterogeneous web data sources.

Contents

CHAPTER 1

Introduction

The intensified use of digital devices, e.g., laptops, tablets and mobile phones, results in an increasing digitization of people's activities. These digital activities generate vast amounts of information about different entities from all sorts of knowledge domains, e.g., education, healthcare, e-commerce, or marketing. The Web has become an ideal place to share and access such information. However, this information is spread over several segments of the Web, such as the Social Web, where we can find profiles of people and organizations, or the Deep Web, where we can find product offers on e-commerce platforms. This segmentation of the Web makes knowledge retrieval and integration a challenging task.

The more the amount of information grows on the Web, the more important is it to develop efficient and effective techniques to search, integrate, and explore this distributed information. Both, academia and industry, research innovative ways to create valuable knowledge out of the information on the Web. Large companies, such as Google, spend a vast amount of resources structuring disparate data from the Web into actionable knowledge. As a result, proprietary knowledge graphs that describe real-world entities and their interrelations are created, e.g., the Google Knowledge Graph[1], the Airbnb Knowledge Graph[2], and the Industrial Knowledge Graph[3] developed by Siemens. However, small and medium-sized organizations, e.g., law enforcement agencies, startups or research institutes, cannot invest comparable resources to create and maintain these knowledge graphs. This work is devoted to easing a knowledge retrieval and integration approach for distributed information spaces on the Web. In the following section, we illustrate the main problem and challenges of this thesis with a motivational example.

1.1 Motivation

Recent studies show that the dominant search task on the Web is the quest for knowledge about entities [2], i.e., about 70% of the web search queries contain one or more entities [3]. This statistic shows that people need to retrieve knowledge about entities from web sources [1]. We see the necessity for searching and integrating knowledge about entities from heterogeneous web sources not only for people but for organizations as well. Consider as a motivational example a case of a journalist who wants to know about the political career of a politically exposed person[4]

[1] https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html

[2] https://medium.com/airbnb-engineering/scaling-knowledge-access-and-retrieval-at-airbnb-665b6ba21e95

[3] https://www.sigs-datacom.de/ots/2018/ki/1-anwendungsszenarien-fuer-wissensnetze-bei-siemens.html

[4] https://en.wikipedia.org/wiki/Politically_exposed_person

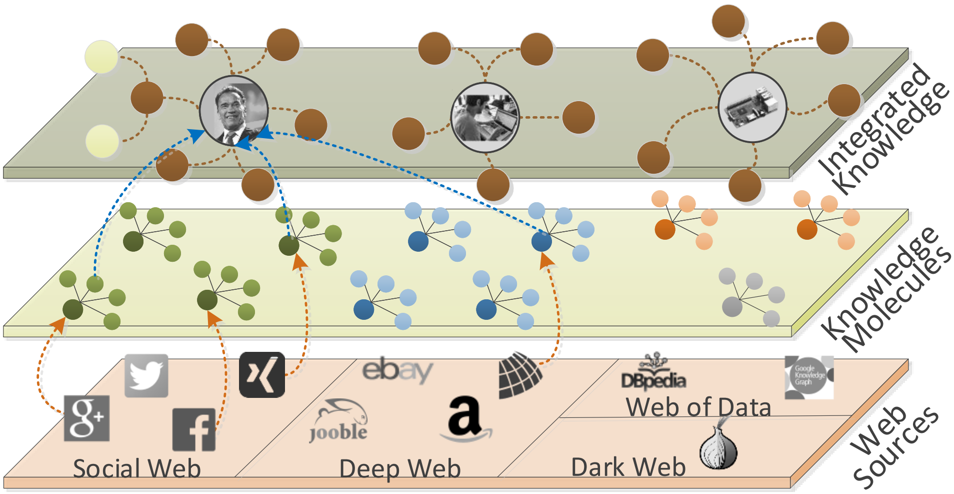


Figure 1.1: **Motivation:** The knowledge about an entity, e.g., a politically exposed person, is spread over different web sources in heterogeneous web segments (Layer 1). There is the need for knowledge representation of these pieces of information (Layer 2), to finally integrate them into a consolidated knowledge graph to get insights about the entity (Layer 3).

and a possible relationship with an offshore company. Figure 1.1 shows how the knowledge about the entities of interest is spread in different segments of the Web (Layer 1 - Web Sources). The typical process for a journalist is to use a Web search engine and start collecting the required bits of knowledge through individual keyword searches (Layer 2 - Knowledge Molecules), in order to finally manually produce integrated and consolidated knowledge about a politician. However, traditional search engines limit us in most cases, e.g., to search for personal profiles on the Web. Traditional search engines fail to search for connections between people and organization mainly because they are limited to only one segment of the Web (the Web of Documents) and do not combine and integrate knowledge from different sources.

Figure 1.1 motivates the need to integrate pieces of knowledge about an entity, e.g., a politician, from heterogeneous web sources. The information about a politician might be spread across social networks, such as Twitter and Facebook, but as well in private catalogs of the Deep Web, such as the OCCRP[5] web source—a journalist association that collects documents related to politically exposed persons. A similar knowledge retrieval and integration scenario can be found in various domains, such as education where the integration of open educational material is needed. Another example is market analysis where a unified view of product offers from different marketplaces is required, e.g., for price comparison in online e-commerce platforms, or exploration of illegal online markets for law enforcement.

Information Retrieval (IR) is a long established research field to search in unstructured data, e.g., HTML websites, and semi-structured data, i.e., XML documents. Several IR solutions, such as Apache Solr[6] and Elastic Search[7] are now driving large-scale information retrieval applications. Likewise, the Semantic Web community has proposed several approaches and platforms, e.g., [4, 5] to provide a unified search across unstructured and semi-structured data. However, for many scenarios and applications, heterogeneous information represented in different

[5] https://www.occrp.org/en

[6] http://lucene.apache.org/solr/

[7] https://www.elastic.co/

modalities (structured, semi-structured, or unstructured) and spread across distributed web sources that have to be made searchable and explorable for end users in an integrated way.

In terms of web sources, the main goal of this thesis is to *retrieve and integrate* knowledge from: the *Social Web*, comprising user-generated content and profiles, e.g., Facebook, Google+, or Twitter social networks; the *Deep Web* comprising Web APIs and data hidden in databases e.g. behind e-commerce platforms, such as eBay and Amazon; the *Web of Data*, with open knowledge bases comprising billions of machine-comprehensible facts as background knowledge; finally, the *Dark Web*, hosting web sources only accessible through specific software, configuration, and authorization mechanisms. The problem of consolidating knowledge about entities from heterogeneous data sources is hard to solve. Many challenges need to be addressed to produce an integrated knowledge asset, e.g., about a politically exposed person. In the following section, we discuss the main problem and challenges motivating this thesis.

1.2 Problem Specification and Challenges

The decentralized and autonomous nature of the Web allows for multiple representations of the same entity, e.g., a politically exposed person. At the conceptual level, we face a knowledge retrieval and integration problem, i.e., "search and integrate pieces of knowledge about the same entity spread on web sources from different segments of the Web". To achieve such knowledge integration several challenges need to be overcome. Figure 1.2 illustrates the three main cross-layer challenges motivating this thesis and preventing us from producing integrated knowledge from heterogeneous web sources.

1.2.1 Challenge 1: Representing Pieces of Knowledge from Web Sources

The first challenge to overcome is the representation of pieces of knowledge spread over heterogeneous web sources. Data can be represented in different levels of structuredness, e.g., structured, semi-structured, and unstructured—*the structuredness conflict* [6], and web sources provide information in all these three levels of structuredness. For structured data, web sources provide Web APIs with a fixed entity model, e.g., the Twitter API to search for user accounts[8]. For semi-structured data, we find web sources containing RDF datasets mainly located in the Web of Data, e.g., the Linked-Leaks dataset[9]. Finally, web sources provide unstructured data in various formats: textual, such as posts in social networks; images, for example, product descriptions in e-commerce sites; or videos shared by users on content platforms.

The web sources are produced, kept, and managed by different organizations using diverse schemata, e.g., Twitter uses the term `User`, while Facebook uses `People` to describe personal information—*the schematic conflict* [6]. This problem is exacerbated by the use of different representations for the same data, for example, different scales or units, various values of precision, different criteria for identifiers, and various encoding methods. Last but not least, each web source may be equipped with specific accessibility, search facility, providing different security mechanisms. For example, Twitter uses application authentication for access to the information, while Facebook requires a user token. Thus, to produce integrated knowledge about entities from heterogeneous web sources, we need a unified knowledge representation that is able to deal with the structuredness, schematic, and accessibility conflicts.

[8] https://developer.twitter.com/en/docs/accounts-and-users/follow-search-get-users/api-reference/get-users-search

[9] https://data.ontotext.com

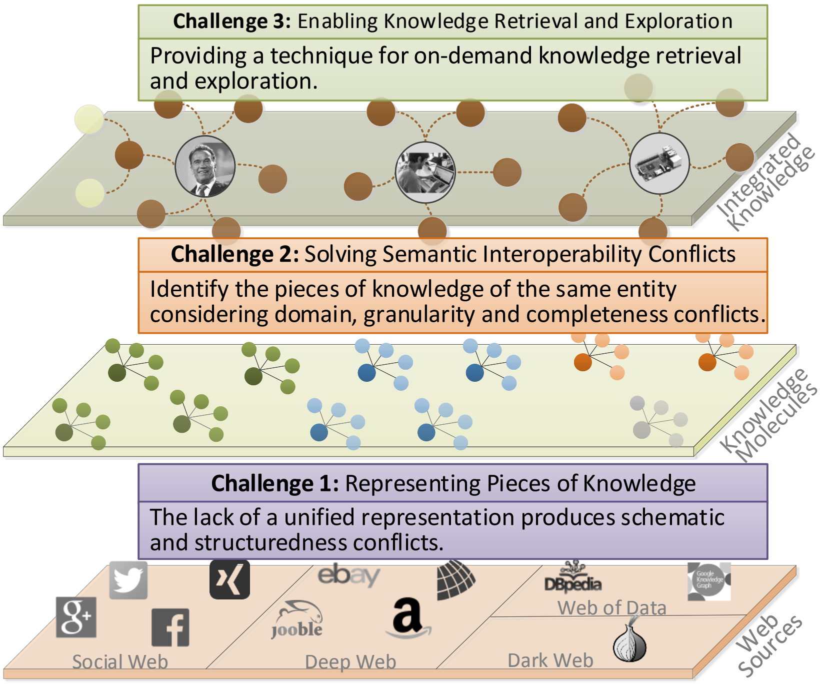

Challenge 3: Enabling Knowledge Retrieval and Exploration
Providing a technique for on-demand knowledge retrieval and exploration.
Integrated Knowledge
Challenge 2: Solving Semantic Interoperability Conflicts
Identify the pieces of knowledge of the same entity considering domain, granularity and completeness conflicts.
Knowledge Molecules
Challenge 1: Representing Pieces of Knowledge
The lack of a unified representation produces schematic and structuredness conflicts.
ebay
DBpedia
Web of Data
jooble
Social Web
Deep Web
Dark Web
Web Sources

Figure 1.2: **Challenges:** To produce integrated knowledge from distributed web sources about entities, we need to solve three main challenges i.e. **(CH1)** Represent pieces of knowledge spread over the web, **(CH2)** Resolve interoperability conflicts at integration time, and **(CH3)** Facilitate knowledge retrieval and exploration on-demand.

1.2.2 Challenge 2: Solving Semantic Interoperability Conflicts

Once the data has been transformed into a homogenous model, the main challenge is to integrate the entities that, albeit described differently, correspond to the same entity. In consequence, semantic interoperability conflicts present on data coming from different web sources need to be solved at integration time. We identify three main semantic interoperability conflicts that need to be solved. The *domain conflict* [6] occurs when various interpretations of the same domain are represented. Different interpretations include: *Homonyms*, the same name is used to represent concepts with a different meaning; *Synonyms*, distinct names are used to model the same concept; *Acronyms*, different abbreviations for the same concept; and *Semantic constraint*, different integrity constraints are used to model the characteristics of a concept. The *granularity problem*, web sources can contain measurements observed at different time-frequency, various criteria of aggregation, and model data at various levels of detail. For example, the Governor of California's current location in Twitter may say Monterey, while his Google+ account indicates just California, meaning that conflicts can occur even because one web source is more precise than another. Finally, web sources usually contain complementary information—the *completeness conflict* [6]. For instance, Twitter and OCCRP[10] contain complementary information about Arnold Schwarzenegger. The integration of such complementary information is required to obtain properly consolidated knowledge about Arnold Schwarzenegger. Thus, we need a semantic integration approach that solves these interoperability conflicts.

[10] https://www.occrp.org/en

1.2.3 Challenge 3: Enabling effective Knowledge Retrieval and Exploration

There exist numerous paradigms to knowledge retrieval and exploration from web sources. However, most current approaches follow a costly Extraction-Transformation-Load (ETL) pipeline. An ETL approach requires access to the entire web dataset to materialize an index which serves as a central repository to provide search and exploration functionality. ETL is an expensive process that not all organizations can afford, and having access to the entire dataset is not possible in many application scenarios [1], e.g., web sources in the Deep Web or personal data. Therefore, the challenge here is to provide a technique for knowledge retrieval and exploration on-demand. A technique oriented on average users (users without expertise in data integration) looking for knowledge on the Web. Additionally, two aspects need to be considered while working with web sources: 1) Web sources are rarely static in their life time, their information changes over time, e.g., the address of a person. The schemata describing the information evolve as well, new relations, new concepts are added or deleted on time. A user may just want to see the latest status of the information or explore the evolution (history) of the information. 2) An integrated knowledge graph does not provide any value if the user cannot find relevant insights by exploring the entities. Users immerse into entities of interest, meandering from topic to topic, exploring for insights. Therefore, a knowledge retrieval approach is needed, an approach that allows users to search entities from a federation of web sources, providing effective techniques to explore the results.

As the problem of knowledge retrieval and integration is much larger and poses many issues and obstacles in different scenarios, we consider the following challenges and problems out of the scope of this thesis: a formal data quality assessment approach is not applied to the integrated knowledge; neither complex logic rules nor advanced reasoning on web sources is tackled by this thesis; the source selection problem is not addressed in this work; finally, structured query transformation is not studied in detail, since we assume all web sources to provide a keyword query mechanism. Nevertheless, the findings presented in this thesis will also serve as a basis for a future work addressing those challenges.

1.3 Research Questions

Based on the main problem and associated challenges described in the previous section, we formulate four research questions in the scope of this thesis:

RQ1: How can semantics encoded in RDF graphs be exploited during the process of integrating data collected from heterogeneous web sources?

With the objective of answering this question, we investigate state-of-the-art approaches for data integration using semantic technologies. We analyze how semantic interoperability conflicts can be solved by the usage of both: semantic similarity metrics, e.g., GADES [7]; and also fusion policies, i.e., combining equivalent entities to create a unified knowledge representation without duplicates. Particularly, we analyze and evaluate the RDF molecule concept [8] in the context of data integration. In the context, of this research question, we assume that the pieces of information are adhering to a unified representation.

RQ2: How can semantic similarity metrics facilitate the process of integrating data collected from heterogeneous web sources?

To address this question, we evaluate state-of-the-art semantic similarity approaches that can be used as a building block for knowledge integration of web sources. We evaluate and compare the accuracy of the integration process using semantic similarity versus non-semantic similarity metrics. Finally, we study the use of a novel similarity metric coming from the Machine Learning world, i.e., a vector representation of the pieces of knowledge—known as embeddings [9].

RQ3: How can knowledge graphs be populated on-demand with data collected from heterogeneous web sources?

Based on the observation that most of the web sources usually provide a Web API, we evaluate an approach to exploit these Web APIs for knowledge integration. Our hypothesis is that these Web APIs can be used to populate knowledge graphs on-demand. We emphasize the scalability analysis of the approach. To complete the retrieval and exploration cycle, we investigate the interaction design patterns for on-demand knowledge exploration. We investigate how the semantics encoded in the RDF graphs provides a more meaningful exploration approach. In particular, we explore the Faceted Browsing approach and investigate its applicability for the scenario of on-demand knowledge exploration.

RQ4: How does semantic data integration impact the adaptability of knowledge retrieval systems?

To address this question, we select various domain-specific applications and apply the techniques and approaches developed in this thesis. We empirically evaluate to what extend the techniques can be tuned to solve domain-specific knowledge integration problems. Our hypothesis is that the usage of semantic technologies for knowledge integration and retrieval can provide a more adaptive and tailored solution for each domain-specific application.

1.4 Thesis Overview

In this section we present an overview of our main contributions, the research areas investigated by this thesis, the references to scientific publications covering this work, and an overview of the thesis structure.

1.4.1 Contributions

The contributions of this thesis are cross disciplinary involving the Data Integration, Information Retrieval and User Interaction fields. Figure 1.3 shows the four main contributions of this thesis.

1. *RDF Molecule-Based Integration Techniques for Heterogeneous Data.* To solve interoperability conflicts among web sources at integration time we devise MINTE, a novel semantic integration technique. MINTE utilizes semantics encoded in ontologies and defines a two-fold approach for both identifying and fusing semantically equivalent entities—what

we call semantic data integration. MINTE defines RDF molecules as the basic unit of the data integration process. We demonstrate two main properties of MINTE i.e., the high adaptability of the approach and the low complexity of the MINTE framework. MINTE defines a set of parameters that can be tuned according to the interoperability conflicts, this allows us to see MINTE as a set of techniques to synthesize knowledge graphs. Empirical evaluations demonstrate the effectiveness of MINTE for the integration of heterogeneous web sources, the experiments use generic and domain-specific datasets. The MINTE approach contributes to answering research question **RQ1**.

2. *A Semantic Similarity Framework for Knowledge Integration.* A similarity metric is a core building block of the MINTE integration approach. Thus, we present a framework that contains a set of similarity metrics that work on RDF molecules for knowledge integration. After evaluating state-of-the-art metrics from the Semantic Web community, we select and adapt GADES [7] a graph-based semantic similarity measure. Moreover, we propose a novel similarity metric for RDF molecules based on embeddings—an embedding is a mapping from RDF molecules to vectors of real numbers, each vector characterize each RDF molecule according to a specific criterion, e.g., its metadata. As a result, we present MateTee, a semantic similarity measure that combines the gradient descent optimization method with semantics encoded in ontologies. MateTee precisely computes values of similarity between RDF molecules, with the advantage that background domain knowledge is not required. We empirically study the accuracy of the similarity framework on the data integration task. The observed results show the benefits of semantic similarity metrics in terms of accuracy with respect to non-semantic methods, these results allow us to answer research question **RQ2**.

3. *A Federated Semantic Search Engine for Web Sources.* Based on our the RDF molecule integration approach that utilizes semantic similarity metrics to integrate pieces of information, we propose FuhSen, a federated semantic search engine. FuhSen is able to create a knowledge graph on-demand from heterogeneous web sources by using their Web APIs. An empirical evaluation of the quality of the FuhSen search engine indicates that FuhSen's approach accurately integrates RDF molecules collected from web sources. Moreover, FuhSen provides a user interface UI adapted for an RDF molecule faceted browsing experience. An evaluation of the usability of FuhSen UI suggest that FuhSen is advantageous compared to purely keyword-based search. The FuhSen approach contributes to answering research question **RQ3**.

4. *An Production-Ready Pipeline to Synthesize Knowledge Graphs from Web Sources* During the development of this thesis, we helped to solve three real-world domain-specific knowledge retrieval and integration scenarios. First, law enforcement agencies needed to collect knowledge about suspects and illegal products (complying to data privacy regulation) from social networks, darknet sites, or on specific web sources in the Deep Web. In the second application, we created a consolidated view of the data scientist job market (at the Europe Union level) by integrating job ads from different job portals. Finally, the third application allows the on-demand completion of knowledge a manufacturing company has about the providers in the supply chain. This knowledge completion enables the company to provide a better experience to their employees by providing additional facts about their providers. Therefore, we implement a production-ready pipeline that includes the MINTE integration approach, the Semantic Similarity Framework, and the FuhSen federated engine. The

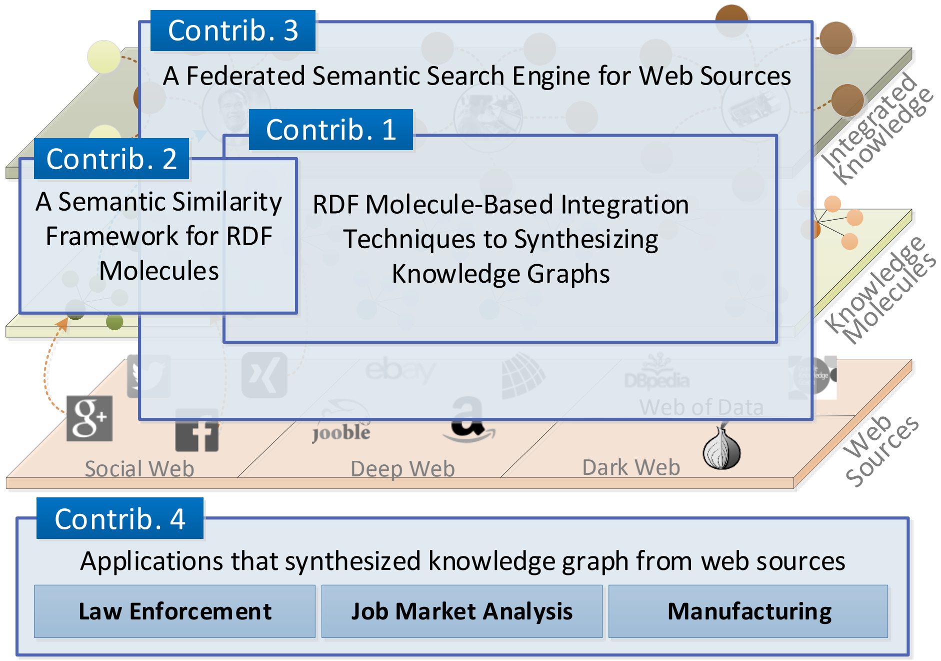

Contrib. 3
A Federated Semantic Search Engine for Web Sources
Contrib. 1
RDF Molecule-Based Integration Techniques to Synthesizing Knowledge Graphs
Contrib. 2
A Semantic Similarity Framework for RDF Molecules
Integrated Knowledge
Knowledge Molecules
Web Sources
Social Web
Deep Web
Dark Web
Web of Data
Contrib. 4
Applications that synthesized knowledge graph from web sources
Law Enforcement
Job Market Analysis
Manufacturing

Figure 1.3: **Contributions:** Four are the main contributions of this thesis including: (1) a novel semantic integration technique; (2) a set of semantic similarity metrics for knowledge integration; (3) a federated search engine to build and explore knowledge graphs on-demand; and (4) the application of the thesis results in three different domain-specific applications.

use of Semantic Technologies allows us to quickly adapt the pipeline to the challenges of each domain-specific application, thus allowing us to answer research question **RQ4**. The three applications are either under pre-production evaluation or in production showing the maturity of the work presented in this thesis.

1.4.2 List of Publications

The work on this thesis has led to multiple scientific publications. Appendix A contains the complete list of publications. In particular, the thesis is based on the following scientific publications (explanations of the authors contributions to these are added for joint-publications with other PhD candidates):

- *Conference Papers*:
  1. **Diego Collarana**, Mikhail Galkin, Christoph Lange, Simon Scerri, Sören Auer, Maria-Esther Vidal. *Synthesizing knowledge graphs from web sources with MINTE$^+$*. In Proceedings of the 17th International Semantic Web Conference (ISWC'18), 359-375;
  2. **Diego Collarana**, Mikhail Galkin, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, Sören Auer. *Semantic Data Integration for Knowledge Graph Construction at Query Time.* In Proceedings of the 11th IEEE International Conference on Semantic Computing (ICSC'17), 109-116;

3. **Diego Collarana**, Mikhail Galkin, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, Sören Auer. *MINTE: semantically integrating RDF graphs.* In Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics (WIMS'17), 22:1-22:11; This is a joint work with Mikhail Galkin, another PhD student at the University of Bonn. In this paper, my contributions include preparing a motivating example, problem and approach definition, architecture of the approach, formalization of fusion policies, preparation of datasets for experiments.

4. Mikhail Galkin, **Diego Collarana**, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, Sören Auer. *SJoin: A Semantic Join Operator to Integrate Heterogeneous RDF Graphs.* In Proceedings of the 28th International Conference of Database and Expert Systems Applications (DEXA'17), 206-221; This is a joint work with Mikhail Galkin, another PhD student at the University of Bonn. In this paper, I contributed to the definition and implementation of the semantic join operators, preparation of datasets for experiments, evaluation, and analysis of obtained results.

5. Camilo Morales, **Diego Collarana**, Maria-Esther Vidal, Sören Auer. *MateTee: A Semantic Similarity Metric Based on Translation Embeddings for Knowledge Graphs.* In Proceedings of the 17th International Conference of Web Engineering (ICWE'17), 246-263; **Best Paper Award.** This is a joint work with Camilo Morales, a Master student at the University of Bonn. I mentored the development of the whole work. In particular, for the article, I contributed to the motivation, the definition, and implementation of the similarity metric, preparation of datasets for experiments, evaluation, and analysis of obtained results.

6. **Diego Collarana**, Mikhail Galkin, Christoph Lange, Irlán Grangel-González, Maria-Esther Vidal, Sören Auer. *FuhSen: A Federated Hybrid Search Engine for Building a Knowledge Graph On-Demand (Short Paper).* In Proceedings of the On the Move to Meaningful Internet Systems OTM 2016 Conferences - Confederated International Conferences CoopIS, CTC, and ODBASE (ODBASE'16), 752-761.

- *Workshops, Demos, and Doctoral Consortium*:

7. **Diego Collarana**, Mikhail Galkin, Maria-Esther Vidal, Mayesha Tasnim. *Synthesizing Data Scientist Job Offers with MINTE$^+$* Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC), 2018;

8. Luis Fuenmayor, **Diego Collarana**, Steffen Lohmann, Sören Auer. *FaRBIE: A Faceted Reactive Browsing Interface for Multi RDF Knowledge Graph Exploration.* In Proceedings of the Proceedings of the Third International Workshop on Visualization and Interaction for Ontologies and Linked Data (VOILA'17), 111-122; This is a joint work with Luis Fuenmayor, a Master student at the University of Aachen. I mentored the development of the whole work. In particular, for the article, I contributed to the motivating example, the definition and implementation of the approach, and analysis of obtained results.

9. **Diego Collarana**. *A Semantic Integration Approach for Building Knowledge Graphs On-Demand.* In Proceedings of the 17th International Conference of Web Engineering (ICWE'17), 575-583;

10. **Diego Collarana**, Christoph Lange, Sören Auer. *FuhSen: A Platform for Federated, RDF-based Hybrid Search.* In Proceedings of the 25th International Conference on World Wide Web (WWW'16), 171-174.

1.5 Thesis Structure

The remainder of this thesis comprises seven chapters organized as follows. Chapter 2 introduces concepts and preliminaries required for the reader. We begin the chapter by describing the segments of the Web and information retrieval techniques to search within them. Next, we introduce the concept of RDF Knowledge Graphs as the foundation for data integration used in this thesis. Finally, we discuss the principles of data integration using semantic technologies. In Chapter 3, we present state-of-the-art related to this thesis. Firstly, we give a complete view of data integration approaches using semantic web technologies. Secondly, we discuss techniques and frameworks to measure the similarity among entities. Finally, we show the most recent approaches for information retrieval and exploration on the Web. Chapter 4 presents a novel semantic integration approach developed in the scope of this thesis. We show how the RDF molecule concept serves as data integration unit, and we describe the use of semantics at each integration step. In Chapter 5, we present a similarity metrics framework including two semantic similarity metrics for knowledge integration, i.e., an adapted version of GADES to work on RDF molecules, and MateTee a semantic similarity metric based on embeddings. We show the results of a detailed performance evaluation of different similarity metrics on the knowledge integration task. Chapter 6 delves into knowledge retrieval and exploration, and we present a novel search engine named FuhSen, which is able to search and integrate knowledge from web sources. FuhSen is defined on top of our integration approach (Chapter 4) and similarity metrics framework (Chapter 5). Additionally, FuhSen defines the properties of a semantic federated search engine capable of building knowledge graphs on-demand from heterogeneous web sources. In Chapter 7, we describe how the approaches defined in this thesis are applied in three domain-specific applications. The applications correspond to both research and industrial projects including law enforcement, job market analysis, and manufacturing scenarios. The implementation is open source and can be used by different research communities and organizations. Chapter 8 finalizes the thesis with a summary of the main results and contributions to the problem of knowledge retrieval and integration from heterogeneous web sources. To conclude the thesis, we define the possible future directions for subsequent research work.

CHAPTER 2

Foundations and Preliminaries

In this chapter, we introduce the concepts and theoretical foundations we will use in later chapters of this thesis. To properly tackle the problem of knowledge retrieval and integration from web sources, the following foundations are required. In Section 2.1, we discuss the main concepts and approaches for Information Retrieval (IR) on the Web. Additionally, we make a clear distinction between a federated versus index-based search engines. Finally, we point out that the keyword search approach is still one of the main approaches for searching on the Web. Section 3.1 presents the foundations of RDF Knowledge Graphs. We introduce RDF Schema as a formal knowledge representation model. Then, one of the core concepts of this thesis is introduced, i.e., RDF molecules. We explain the formal query language for RDF knowledge graphs, i.e., SPARQL. Finally, the most well-known approach for RDF graph exploration is described, i.e., Faceted Browsing. In Section 2.3, we present the principles of data integration using Semantic Technlogies. Firstly, we present the keyword search as a novel on-demand integration approach. Secondly, we introduce the use of RDF as the lingua franca for data integration. Thirdly, the problem of entity matching is introduced as well as the usage of similarity metrics to solve this problem. Lastly, we discuss the semantic interoperability conflicts we need to tackle during the integration process.

2.1 Searching on the Web

2.1.1 Segments of the Web

The accessibility, the growing, and content purpose of the data on the Web has lead to the division of the Web in the following segments: the Web of Documents, containing HTML documents hosted in websites; the Web of Data, comprising billions of machine-comprehensible facts; the Social Web, comprising user-generated content and profiles; the Deep Web, containing websites hidden behind HTTP forms; finally, the Dark Web, containing websites accessible only via the usage of a specialized software, configuration, and authorization. Figure 2.1 shows an image representation of the segmentation of the Web according to its visibility.

The Web of Documents

The Web of Documents is the portion of the Web that is readily available to the general public and searchable via the standard web search engines such as Google [10]. It can be seen as an information space composed mainly by HTML documents. The HTML documents and other

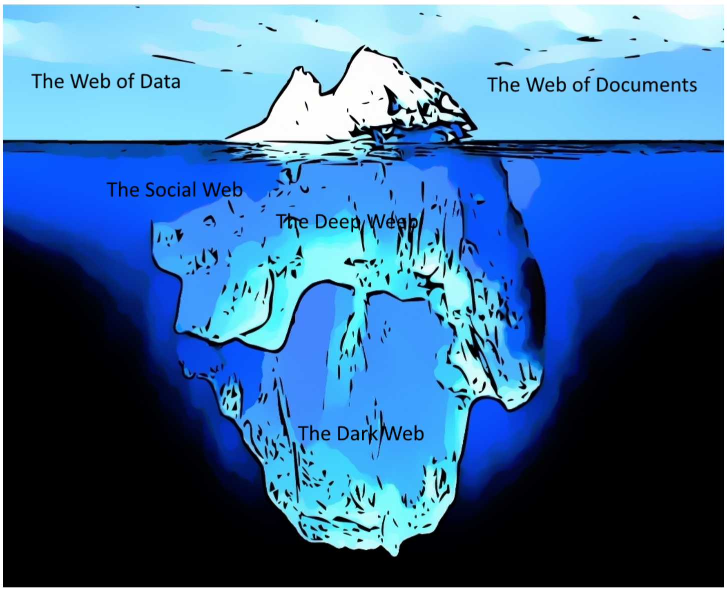


Figure 2.1: **Segments of the Web:** According to its visibility and content, the Web can be conceptually segmented in: The *visible web* containing information that web crawlers can reach, the visible web comprises the Web of Documents and the Web of Data. The *invisible web* contains information that traditional web crawlers cannot reach, it comprises the Social Web, the Deep Web, and the Dark Web.

web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet. The HTML documents can be accessible openly via a Web Browser such as FireFox. By crawling these HTML Documents huge indexes are created by companies such as Google, or Bing. According to the World Wide Web Size[1] organization, the Google index size of the Web of Documents reaches 14.5 billion pages. This index is accessible through API, for example, someone can use Google Custom Search API[2] to search in the Google index. In the recent years, the HTML documents have been annotated with semantic information (RDFa[3]) in order to provide more semantic information its content. The semantics annotations in the websites are then used by crawlers to improve the index creation and the interpretation of the content of the website.

The Web of Data

C. Bizer et al. [11] explain that “Traditionally, data published on the Web of Documents have been made available as raw dumps in formats such as CSV or XML or marked up as HTML tables, sacrificing much of its structure and semantics”. This gives birth to the Web of Data, a web containing structured semantic data that machines can understand. In the Web of Data, we can find datasets comprising billions of machine-comprehensible facts. These datasets provide important background knowledge, e.g., spatial context information for aggregating information. The most well-known datasets in this category are: DBpedia[4] [12], and GeoNames[5] datasets.

[1] http://www.worldwidewebsize.com/

[2] https://developers.google.com/custom-search/json-api/v1/overview

[3] https://www.w3.org/TR/rdfa-primer/

[4] https://wiki.dbpedia.org/

[5] http://www.geonames.org

We refer to the Linked Open Data Cloud[6] (LOD) to have an estimation of the size of the Web of Data. In the latest report, the LOD contains about 1,163 datasets comprising billions of facts. Similarly to the Web of Documents, the accessibility of the Web of Data is open. Datasets on the Web of Data commonly provide SPARQL endpoints as API for querying and exploration.

The Social Web

According to Wikipedia[7], the Social Web is "is a set of social relations that link people through the Web". The Social Web consists of user-generated content, plus connections among people and their objects of interest. Breslin et al. [13] argue that the connections created by people on online social websites are established through social objects of common interest, e.g., the content they create together, co-annotate, or for which they use similar annotations. Therefore, what clearly distinguishes the Social Web is the ability of users to interact with each other via the content published on the social networks. Facebook, Google+ and Twitter are the most relevant social networks. The number of users at 2017 is estimated at 2,46 billion. The user needs to register as a member of a social network for producing and browsing content, it means an account and a password is required. These social networks often provide a REST API interface to access and query their data.

The Deep Web

The Deep Web, also known as the hidden web [14], is the segment of the Web which content is not indexed by standard web search engines. The content of the Deep Web is hidden behind HTML forms and its purpose varies according to its application. Common applications in the Deep Web are webmail, online banking, on-demand video, newspapers, and many more services that users must pay to get access (and which are protected by a paywall). The Deep Web size is estimated to be 500 times bigger than the Web of Documents [15]. Web sources hosted in the Deep Web usually provide a Web API interface for accessing their data. Several projects such DARPA Memex [16] have created an index of provides an API to access it. E-commerce platforms as well provide APIs to query the content, e.g., eBay[8].

The Dark Web

The Dark Web is an overlay network of the Web, accessible only through specific software, configuration, authorization, and often using non-standard communication protocols and ports. Two typical darknet types are friend-to-friend networks (usually used for file sharing with a peer-to-peer connection), and private networks such as Tor. The Dark Web is a subset of the Deep Web, and it is known for hosting illegal activities. The most remarkable example is the Dark Market Silk Road[9], best known as a platform for selling illegal drugs and weapons. In order to access the content, a user needs specific software such as Tor Browser and Tor Proxy[10]. Several projects have started to create indexes of the Darknet websites, for example, GRAMS[11] and Onion.city and the information can be accessed through their APIs.

[6] http://lod-cloud.net/

[7] https://en.wikipedia.org/wiki/Social_media

[8] https://go.developer.ebay.com

[9] https://en.wikipedia.org/wiki/Silk_Road_(marketplace)

[10] https://www.torproject.org/projects/torbrowser.html.en

[11] https://de.wikipedia.org/wiki/Grams_(Suchmaschine)

Segment	Content	Size	Accesibility	APIs
1. Web of Documents	HTML Docs	14 billion sites	Open	Google, Bing, etc.
2. Web of Data	RDF	1,163 datasets	Open	SPARQL Endpoints
3. Social Web	User Content	2,46 billions	User Tocken	Facebook, Twitter, etc.
4. Deep Web	HTML Docs	500 times 1	Forms	DARPA, etc.
5. Dark Web	HTML Docs	Deep Web subset	Special Software	GRAMS, Onion.city, etc.

Table 2.1: **Web segments**. Characterization of the web segments relevant for the scope of this thesis.

Table 2.1 shows a characterization of the Web's segments according to their content, size, accessibility, and Web APIs they offer to query their data. We observe in Table 2.1 the variety of the data on the Web. A single approach to create a unique index from these segments of the Web is costly and maybe not even feasible. This observation suggests that a federated approach to retrieve the knowledge from web sources is more convenient. A fact discovered during this thesis is that all of the segments provide Web APIs for searching and integration purposes.

2.1.2 Information Retrieval

The purpose of Information Retrieval (IR) systems is to help people find the right (most useful) information, in the right (most convenient) format, at the right time (when they need it). The main activity of an IR system is obtaining relevant sources to an information need from a collection of information sources. IR is a mature science field of searching for information in documents, searching for documents themselves, but also for searching metadata that describes the documents, as well as for images and sounds. Searches can be based on full-text or other content-based indexing. An IR process begins when a user enters a query into the IR system. Queries are formal statements of information needs, for example, keywords in web search engines. In information retrieval, a query does not uniquely identify a single object in the collection. Instead, several objects may match the query, with different degrees of relevancy. To evaluate the performance of an IR system, i.e., how well a system meets the information needs of its users, the most common evaluation metrics includes precision and recall.

Definition 2.1: Information Retrieval Evaluation Metrics

a) Precision is the fraction of the documents retrieved RD that are relevant to the users information need R.

$$Precision = \frac{|R \cap RD|}{|RD|}$$

b) Recall is the fraction of the documents that are relevant to the query R that are successfully retrieved.

$$Recall = \frac{|R \cap RD|}{|R|}$$

c) F-measure is the weighted harmonic mean of Precision and Recall, the traditional F-measure or balanced F-score is.

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

2.1.3 Federated Search Engines

Two are the main approaches in the IR field: Index-based and Federated-based engines. Shokouhi el al. [17] define federated search as a "technique for searching multiple text collections simultaneously, also know as federated information retrieval". A user makes a single query that is distributed to other search engines participating in the federation—federation members. Then, the federated search engine aggregates the results to create a consolidated view of the results for presentation to the user. At difference of index-based search engines, which create huge indexes of documents by crawling the Web, federated-based search engines produce results on-demand. Shokouhi el al. [17] show that index-based search engines "cannot easily index uncrawlable hidden web collections while federated search systems can search the contents of hidden web collections without crawling". Three are the main challenges for federated-based search engines: the source selection problem, for each query the most suitable federation members need to be selected; the source representation problem, we need to specify the type of results each federation member can provide; and the merging problem, the results returned from each federation member need to be merged before the final presentation to the user.

2.1.4 Keyword Search

Keyword search is still one of the main user interfaces approaches to retrieve information from web sources [18]. Tran et al. [19] reported that "keyword queries enjoy widespread usage as they represent an intuitive way of specifying information needs". The keyword search approaches fit well the scenarios where the users search for knowledge about entities spread on web sources [1]. In traditional information retrieval, a keyword search retrieves a ranked list of documents with matches to all of the keywords. A keyword query consists of a set of terms, each term gets matched against document's content, and the highest-scoring matches are returned.

Doan et al. [20] showed that keyword search approaches are also used to search on structured data, such as relational databases; and semi-structured data, such as XML where the goal is to find different nodes that match the keywords. To answer keyword queries over structured and semi-structured data, the general approach is representing the data as a graph relating data and metadata items. Nodes in the graph represent attribute values—and in some cases metadata items such as attribute labels or relations. Directed edges represent conceptual links between the nodes, e.g., foreign keys in relational databases. Then, the keyword query gets matched against the node in the graph, and the highest-scoring nodes are returned as answers.

Although data sets on the Web of Data are equipped with its own query language, i.e., SPARQL, keyword search is still supported [18]. All the main dataset stores, supporting the Web of Data, such as Virtuoso or Jena Fuseki support keyword search. [12] The main reason is that to write SPARQL queries a user requires some expertise in graph patterns, which limits the accessibility to a broad scope of users.

2.2 RDF Knowledge Graphs

The term Knowledge Graph was coined by Google in 2012[13] as a novel knowledge management paradigm. The concept received a significant attention in the research community, especially in the Semantic Web community. Thus, several public and private knowledge graphs (e.g.,

[12] https://jena.apache.org/documentation/query/text-query.html

[13] https://www.blog.google/products/search/introducing-knowledge-graph-things-not/

DBpedia [21], Wikidata [22], Yahoo [23], Microsoft[14], Facebook[15]) have been developed to support the provision of smart services. For example, knowledge graphs enable web search engines to search for entities, e.g., people or places stored in knowledge graphs and instantly get information that's relevant to a user's query. The prime source of information to build a knowledge graph is the Web, containing data from different domains, e.g., government, scientific communities, social media, etc. Although there is not an agreement upon a formal definition of knowledge graphs, in this thesis we use the definition presented by Paulheim in 2017:

Definition 2.2: Knowledge Graph [24]

1. mainly describes real world entities and their interrelations, organized in a graph;
2. defines possible classes and relations of entities in a schema;
3. allows for potentially interrelating arbitrary entities with each other;
4. covers various topical domains.

A data model is required to build a knowledge graph under the definition presented by Paulheim. The Resource Description Framework[16] (RDF) is a data model to describe resources on the Web of Data. It is part of specification family driven by the Semantic Web research community and the W3C[17]. Figure 2.2 shows the semantic web technology stack where we can observe technologies to query, exchange, represent and format data on the Web. RDF is the W3C recommended standard for exchanging data on the Web, and it is the perfect data model for building knowledge graphs. RDF as a data model is capable of using a variety of syntax, notations, and data serialization formats. A large amount of data has been converted to RDF, often as multiple datasets physically distributed over different locations.

The basic structure of RDF is the triple: subject, predicate, object. Based on this basic structure big knowledge graphs have been built, e.g., Wikidata [22] and DBpedia [21]. These knowledge graphs have become powerful assets for enhancing search, and they are being intensively used in both academia and industry. Although it is difficult to measure the value of a knowledge graph, it serves as a basis for empowering enterprise information applications such as Semantic Search Engine, Entity Recognition, Question Answering Systems or Data Integration. RDF knowledge graphs help to automatically solve data-driven oriented tasks, providing more useful and meaningful services from heterogeneous data [26] such as web sources.

Definition 2.3: RDF Triple [27]

Let $\mathbf{I}$, $\mathbf{B}$, $\mathbf{L}$ be disjoint infinite sets of URIs, blank nodes, and literals, respectively. A tuple $(s, p, o) \in (\mathbf{I} \cup \mathbf{B}) \times \mathbf{I} \times (\mathbf{I} \cup \mathbf{B} \cup \mathbf{L})$ is denominated an RDF triple, where s is called the subject, p the predicate, and o the object.

[14] https://blogs.bing.com/search/2013/03/21/understand-your-world-with-bing/

[15] https://www.facebook.com/notes/facebook-engineering/under-the-hood-the-entities-graph/10151490531588920

[16] https://www.w3.org/RDF/

[17] https://www.w3.org/standards/semanticweb/

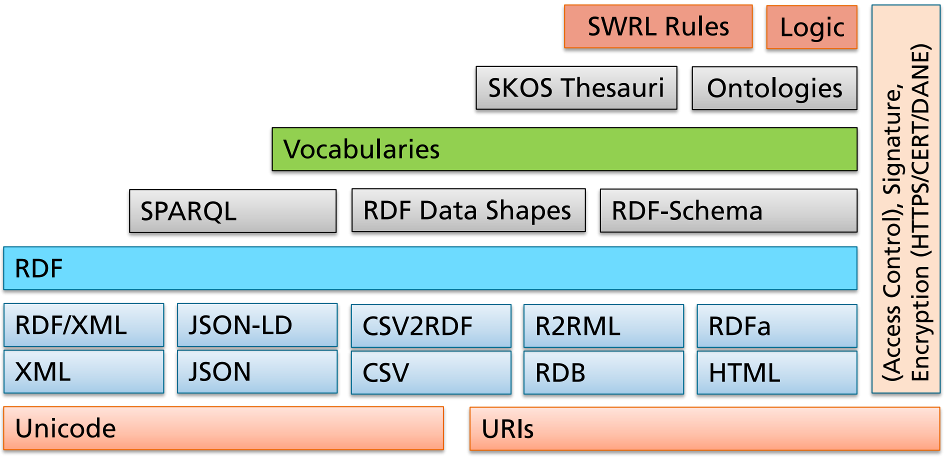


Figure 2.2: The Semantic Web Layer Cake 2015 – Bridging between Big and Smart Data. Source of picture [25]

2.2.1 RDF Schema

In order to add more semantic information to the data, the RDF data model relies on schemas—RDFS and OWL. These schemas enrich the resources such that computer algorithms can make sense out of them. RDF Schema[18] (RDFS) is an extension to RDF and it provides a data-modeling vocabulary RDF data. RDFS provides meaning to things described as RDF entities, e.g., what is an athlete, what is a politician, what is the relation between them. RDFS allows to model not have only nodes and edges (Values), but meaning as well (Schema). RDFS allows defining constraints, the type, and characteristics of an entity of interest. RDFS allows to model hierarchies of classes and properties, which provides meanings for reasoning. It is the simples modeling languages in the Semantic Web Technology Stack. The RDFS most important concepts are: `rdfs:Class` and `rdfs:subClassOf`, enabling hierarchical classes structures; `rdfs:subPropertyOf`, enabling hierarchical properties structures; `rdfs:domain` and `rdfs:range`, allow to identifying the type of the subject and the type of the object value of a triple; finally, `rdfs:comment` and `rdfs:label`, allows to add human-readable annotations. Figure 2.3 illustrates an example RDF knowledge graph of a politically exposed person using RDF and RDFS schemas. The graph contains all RDF classes (e.g., ex:Person) as well as instances (e.g. ex:Arnold-Schwarzenegger), literals (e.g., a location California).

Definition 2.4: Triple Pattern [28]

Let U, B, L be disjoint infinite sets of URIs, blank nodes, and literals, respectively. Let V be a set of variables such that $V \cap (U \cup B \cup L) = \theta$. A triple pattern *tp* is member of the set $(U \cup V) \times (U \cup V) \times (U \cup L \cup V)$. Let $tp_1, tp_2, \ldots, tp_n$ be triple patterns. A Basic Graph Pattern (BGP) B is the conjunction of triple patterns, i.e., $B = tp_1 AND tp_2 AND \ldots AND tp_n$

[18] https://www.w3.org/TR/rdf-schema/

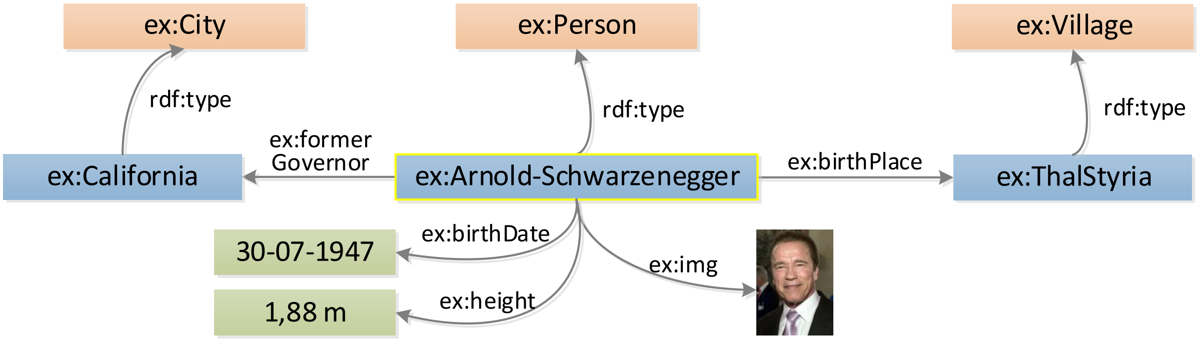


Figure 2.3: **RDF Knowledge Graph:** Excerpt of an RDF knowledge graph describing the RDF molecule of a politically exposed person.

2.2.2 RDF Molecule

To manage the knowledge of a specific real-world entity in a knowledge graph, a finer grain unit is the concept of RDF molecule. While an RDF knowledge graph defines the knowledge of a whole domain, the RDF molecule is bounded to a single entity. Figure 2.3 shows the RDF molecule of a politically exposed person. The RDF molecule of links all the knowledge regarding to the politician such as the birth place, the birth data, or the height. Thus, all the statements have as suject `ex:Arnold-Schwarzenegger`. Formally, an RDF molecule is defined as follows:

Definition 2.5: RDF molecule [8]

Given an RDF graph G, an RDF subject-molecule $M \subseteq G$ is a set of triples $t_1, t_2, \ldots, t_n$ in which $subject(t_1) = subject(t_2) = \cdots = subject(t_n)$.

2.2.3 RDF Query Language

SPARQL[19] is the W3C recommend language to query RDF datasets. SPARQL is able to retrieve and manipulate data stored in RDF format. It is a W3C standard recognized as one of the key technologies of the Semantic Web. The latest version is SPARQL 1.1 released in March 2013. A SPARQL query consists of triple patterns, conjunctions, disjunctions, and optional patterns. Triple patterns are similar to RDF triples where the subject, predicate, or object may be variables. In a query, variables act like placeholders which are bound with RDF terms to build the solutions of the query. The expressive power of SPARQL [29] comes in the ability to combine data properties and the schema of the data, it consists of five parts:

- **Prefix Declaration:** a list of URI prefixes to avoid writing complete URIs in the query.
- **Dataset Clause:** similarly to SQL databases, where the user specifies the schema to be used, in the dataset clause is specified which graph is going to be queried.
- **Result Clause:** in this clause the type of query (SELECT, ASK, CONSTRUCT or DESCRIBE) and the variables to return are specified.

[19] https://www.w3.org/TR/rdf-sparql-query/

- **Query Clause:** it contains the patterns that have to be matched in the graph. Resources fulfilling the specified patterns will be associated with the corresponding variables in the result clause.
- **Solution Modifiers:** the results of the queries can be paginated, ordered or sliced.

Listing 2.1 and 2.2 show examples of SELECT and CONSTRUCT queries on the RDF knowledge graph that describes knowledge about a politician. The query request the name of the people who has been governor of the city of California.

```
PREFIX ex: <http://example.org/2017/03/schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?name
WHERE {
  ?s rdf:type ex:Person .
  ?s ex:formerGovernor ex:California .
  ?s ex:name ?name
}
```

Listing 2.1: Select SPARQL query example

```
PREFIX ex: <http://example.org/2017/03/schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

CONSTRUCT
?s rdf:label ?name .
?s rdf:type ?Politician .
WHERE {
?s rdf:type ex:Person .
?s ex:formerGovernor ex:California .
?s ex:name ?name
}
```

Listing 2.2: Construct SPARQL query example

2.2.4 RDF Graph Exploration

RDF knowledge graphs such as DBpedia, Yago, or Freebase have become a powerful asset for many applications, and they are being intensively used in both academia and industry [30]. One of the main applications supported by RDF graphs is knowledge exploration since the RDF data model is ideal for performing an explorative search. Accordingly, White et al. [31] depict it as a scenario: "typified by uncertainty about the space being searched and the nature of the problem that motivates the search", where the following situations may occur. Firstly, the search target is either fully, or partially, unknown. Secondly, the search begins with a given degree of certainty about known facts, which evolve into unknown and unfamiliar upon exposure to new information. Lastly, the users distinguish valuable information portions by scanning through different resources, where they evaluate their usefulness and eventually determine their content and structure. Knowledge graphs such as DBpedia, Yago, or Freebase are typically oriented towards end-users search, thus a critical challenge is to provide an appropriate user interaction and user interface oriented to end-users. User interaction to explore RDF knowledge graphs attracted a great deal of attention in the Semantic Web community, the goal is to develop of simple yet powerful query interfaces for non-expert users [32–34].

Faceted Browsing

Faceted search is the de-facto approach for exploratory search in RDF repositories [35]. It has its origins in the e-commerce applications, and it has been shown as a suitable approach for RDF knowledge graph exploration. Faceted search is an approach for querying collections of entities where users can narrow down the search results by applying filters—called facets [36]. A facet typically consists of a predicate (e.g., 'gender' or 'occupation' when querying entities about people) and a set of possible string values (e.g., 'female' or 'research'), and entities in the collection are annotated with predicate-value pairs. During faceted search users iteratively select facet values and the entities annotated according to the selection are returned as the search result. Faceted search of RDF datasets has received significant attention and many approaches have been developed [37–39]. Furthermore, several of those systems have been successfully applied on big knowledge graphs exploration, e.g., exploring Freebase knowledge graph [40].

2.3 Semantic Data Integration

2.3.1 Principles of Data Integration

The main problem of this thesis can be tackled from the Data Integration perspective, i.e., we have heterogeneous data spread over web sources. **Data integration** is the process of combining data from diverse sources and providing a unified view to work with. Data integration systems are formally defined as a triple $< O, S, M >$, where O is the global (or mediated) schema, S is the heterogeneous set of source schemas, and M is a set of mappings between the source and the global schema. A data integration system is formalized as follows:

Definition 2.6: Data Integration [41]

A data integration system IS is defined as a tuple $< O, S, M >$, where

- O is the global schema (e.g., RDF Schema), expressed in a language L_O over an alphabet A_O. The alphabet A_O consists of symbols for each element in O.
- S is the source schema, expressed in a language L_S over an alphabet A_S. The alphabet A_S contains sybmbols for each element of the sources.
- M is the mapping between O and S that is represented as assertions: $q_s \to q_o$; $q_o \to q_a$. Where q_s and q_o are two queries of the same arity, q_s is a query expressed in the source schema, q_o is a query expressed in the global schema. The assertions imply correspondence between global and source concepts.

The data integration process implies the combination of sources with heterogeneous schemas into a unified view, e.g., for knowledge graph construction. One of the most important contributions of the Semantic Web research community is the applicability of the Semantic Technology Stack on the data integration problem. Then, **Semantic Data Integration** is the use of semantic web technologies to solve data integration problems. By using semantic technologies, i.e., the W3C standards RDF, RDFS, OWL, and SPARQL, heterogeneous datasets can be integrated in a more flexible way, since the information about entities and their relationships are held together in a meaningful way. Thus, the process of semantic data integration creates an interrelated information space that facilitates the management of the knowledge derived from the data, providing a 360° view of the data. There are two main approaches to integrate data: Materialized approach, the data is moved to a central repository; and the Virtual approach; the data remain at the source and the integration is performed at query time. Virtual approaches are more suitable for integration scenarios where the data frequently change, and a variety of entities exits in the data sources, which is the case of web sources.

Mediator-Wrapper Architecture

It is a well-known architecture for virtual data integration. *Wrappers* are the components of a data integration system that communicates with the data sources. The task of a wrapper involves sending queries from the higher levels of the data integration system to the sources, converting then the replies to a format that can be manipulated by the *Mediator*. The *Mediator* orchestrates the executing of the wrappers and merge the results into a consolidated view of the data [42]. The complexity of the wrapper depends on the nature of the data source. For example, a wrapper to a web API would translate a query into the appropriate HTTP request. When the answer comes back in JSON format, the wrapper would extract the objects and translated them to a global schema that the mediator knows. There are two main types of mediators in the literature [43]: Local-as-View (LAV) and Global-as-View (GAV).

Local-as-View Mediator

Local-as-View (LAV) mediation [44] is a well-known and flexible approach to perform data integration over heterogeneous and autonomous data sources. A LAV mediator relies on views to define semantic mappings between a uniform interface defined at the mediator level, and local schemas or views that describe the integrated data sources. A LAV mediator employes a query rewriter to translate a mediator query into the union of queries against the local views. LAV is suitable for environments where data frequently change, and entities of different types are defined in a single source. The formal definition of a LAV mediator is the following:

Definition 2.7: Local-as-View [41]

In a data integration system $IS =< O, S, M >$ based on LAV approach the mapping M associates to each element s of the source schema S a query q_o in terms of the global schema O: $s \rightarrow q_o$, i.e., the sources are represented as a view over the global schema.

Keyword Search as Integration on Demand

Traditional data integration objectives are: build applications, whether Web-based or more traditional, that provide cross-source information access; and data analysis plus exploration via sophisticated query interfaces performed by sophisticated users [20]. However, recent attention has been paid to enable "average users" (non-database-experts) to pose ad-hoc queries over integrated data via a familiar interface, e.g., keyword search [45]. Keyword search is a good approach to reduce the complexity of typical data integrated query interfaces. Intuitively, the set of keyword terms describes a set of concepts in which the user is interested. Then, the data integration system task is to find the related tables, objects, or tuples to these concepts.

2.3.2 RDF the Lingua Franca of Data Integration

One of the main challenges in data integration is to provide a global schema O, which is flexible but expressive at the same time. RDF and RDFS provide a mean to create from lightweight vocabularies to heavyweight ontologies containing logical rules for data integration. Frischmuth et al. [46] present RDF as the "Lingua Franca for Data Integration". RDF is simple, less invasive, and different kinds of data models (relational, taxonomic, graphs, object-oriented, etc...) can be easily encoded and combined. RDF support a variety of serializations to interface with other applications, RDF can be serialized as HTML with RDFa, XML with RDF-XML, JSON with JSON-LD, and CSV. Additionally, RDF supports distributed data and schema. Finally, the RDF representational unit, "the triple—subject, predicate, object", facilitates mashing data from different perspectives, i.e., facts, entity-relation, logical axioms, and objects. All these characteristics make RDF a suitable data model for solving complex data integration scenarios, enabling the creation of sustainable data ecosystems.

2.3.3 Entity Matching

Entity matching is the problem of finding data from different sources that refer to the same real-world entity. Entity matching plays a critical role during the data integration process, and it raises two major challenges: accuracy and scalability. Matching entities accurately is difficult because data that refer to the same real-world entity is often very different, e.g., misspelled, different format, incomplete data, etc. Scalability refers to the problem of finding similar entities among a large number of entities, a comparison of all pairs of entities would be quadratic in time and therefore impractical. The entity matching problem is formalized as follows:

Definition 2.8: Entity Matching [20]

Given two sets of entities X and Y, we want to find all pairs of entities (x, y), where $x \in X$ and $y \in Y$, such that x and y refer to the same real-world entity.

Similarity Metric

A similarity metric maps a pair of entities (x, y) into a number in the range [0,1], a higher value indicates greater similarity between x and y. The terms distance and cost have also been used to describe similarity metrics, except that smaller values indicate higher similarity. Values close to 0 indicate that the compared objects are dissimilar while values close to the supreme of I_s correspond to very similar objects. There exist three types of similarity metrics: the distance-based similarity measures, the feature-based similarity measures, and the probabilistic similarity measures [47]. In this thesis, we evaluate distance-based and feature-based similarity metrics to solve the problem of entity matching among web sources.

Definition 2.9: Similarity Metric *Sim* [48]

A similarity metric on elements in X is an upper bound, exhaustive, and total function Sim: $X \times X \rightarrow I_s \subset \Re$ with $| I_s |> 1$. I_s represents a set of real numbers containing at least two elements that allow to distinguish similar from dissimilar elements. Defining the range of a similarity measure as I_s is equivalent to defining a minimum and a maximum value. Thus, I_s is upper and lower bounded and the set has both a supremum (sup) and an infimum (inf).

2.3.4 Data Provenance

Data provenance is a record of the origins of the data, i.e., which operations were applied, where it moves over time, etc. Data provenance gives visibility to trace errors back to the root cause in the data integration process. In the broadest sense, the provenance may include a huge number of factors, e.g., who created the initial data, when was created, or what equipment was used. In the Semantic Web community, data provenance is model commonly using vocabularies such as the provenance ontology[20] (PROV). The PROV ontology provides a set of classes, properties, and restrictions that can be used to represent provenance information generated during the creation and transformation of an RDF entity.

2.3.5 Semantic Interoperability Conflicts

To integrate heterogeneous sources in a unified way, Bellazi et al. [49] show the importance of analyzing the data sources to identify interoperability conflicts. Vidal et al. [6] characterize the interoperability conflicts into six categories. Figure 2.4 summarizes the main characteristics of each interoperability conflict.

1. Structuredness (C1): data sources may be described at different levels of structuredness, i.e., structured, semi-structured, and unstructured. The entities in a structured data source are described in terms of fixed schema and attributes, e.g., the entity-relationship model. In semi-structured data sources, a fixed schema is not required, and entities can be represented using different attributes and properties. Examples of semi-structured data models are the Resource Description Framework (RDF) or XML. Lastly, in unstructured data sources the no data model is used, so the data does not follow any structured. Typically unstructured data formats are: textual, numerical, images, or videos.

2. Schematic (C2): the following conflicts arise when data sources are modeled with different schema. i) the same entity is represented by different attributes; ii) different structures model the same entity, e.g., classes versus properties; iii) the same property is represented with different data types, e.g., string versus integer; iv) different levels of specialization/-generalization describe the same entity; v) the same entity is named differently; and vi)

[20] https://www.w3.org/TR/prov-o/

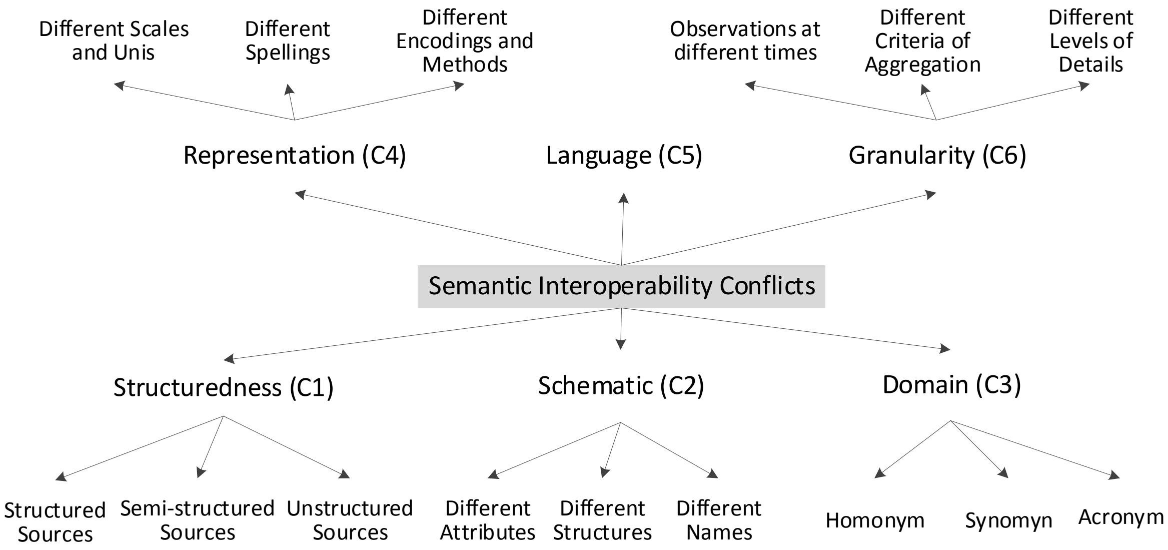


Figure 2.4: **The Semantic Interoperability Conflicts** existing among heterogeneous sources divided into six main categories. Picture based on the book chapter [6]

different ontologies are used, e.g., to describe a gene function the following ontologies may be used UMLS, SNOMED-CT, NCIT, or GO.

3. Domain (C3): various interpretations of the same domain exist on different data sources. These interpretations include: homonyms, synonyms, acronyms, and semantic constraints—different integrity constraints are used to model a concept.
4. Representation (C4): different representations are used to model the same entity. These representation conflicts include: different scales and units, values of precision, incorrect spellings, different identifiers, and various encodings.
5. Language (C5): the data and schema may be specified using different languages.
6. Granularity (C6): the data may be collected under different levels of granularity. Examples of granularity include: samples of the same measurement observed at different time-frequency, various criteria of aggregation, and data model at different levels of detail.

CHAPTER 3

Related Work

This chapter reviews the most relevant state-of-the-art approaches related to this thesis. Different approaches exist in the literature regarding the problem of retrieving and integrating pieces of knowledge about entities spread over web sources. Figure 3.1 shows the dimensions defined during the literature review. For each dimension, we present an overview of the approaches highlighting their limitations to solve the challenges defined in the scope of this thesis. Firstly, in Section 3.1 we discuss the use of semantic technology to solve the problem of heterogeneous data integration. We show the shortcomings of the state-of-the-art techniques on the problem of integrating semantically equivalent entities from heterogeneous web sources. Secondly, Section 3.2 presents a summary of the state-of-the-art metrics to determine the similarity between two entities in an RDF graph. We review a spectrum of methods ranging from classic metrics such as Jaccard to more advanced machine learning metrics using multidimensional vector representation of entities, i.e., embeddings. Finally, Section 3.3 presents the most recent approaches in the Information Retrieval field. We show that these approaches focus only on particular segments of the Web, i.e., search engines for the Deep Web or for the Web of Data. We close the section by showing faceted browsing approaches to explore the results of the search engines, we particularly focus on approaches for knowledge graphs.

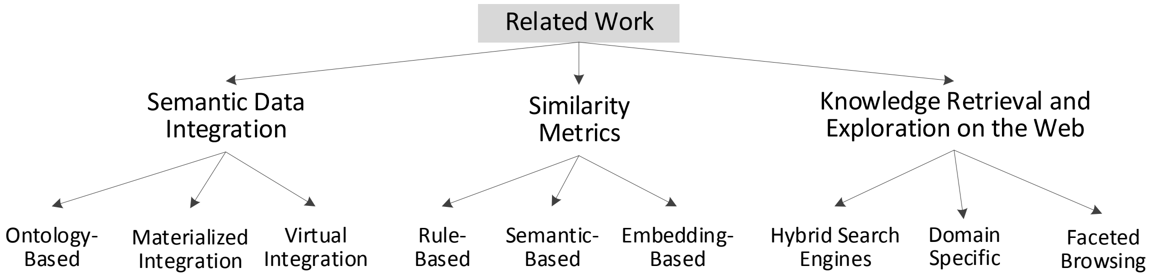


Figure 3.1: **Dimensions of the Related Work:** We present the works related to this thesis in three dimensions including Semantic Data Integration approaches, Similarity Metrics for Entity Matching, and Search Engines for the Web.

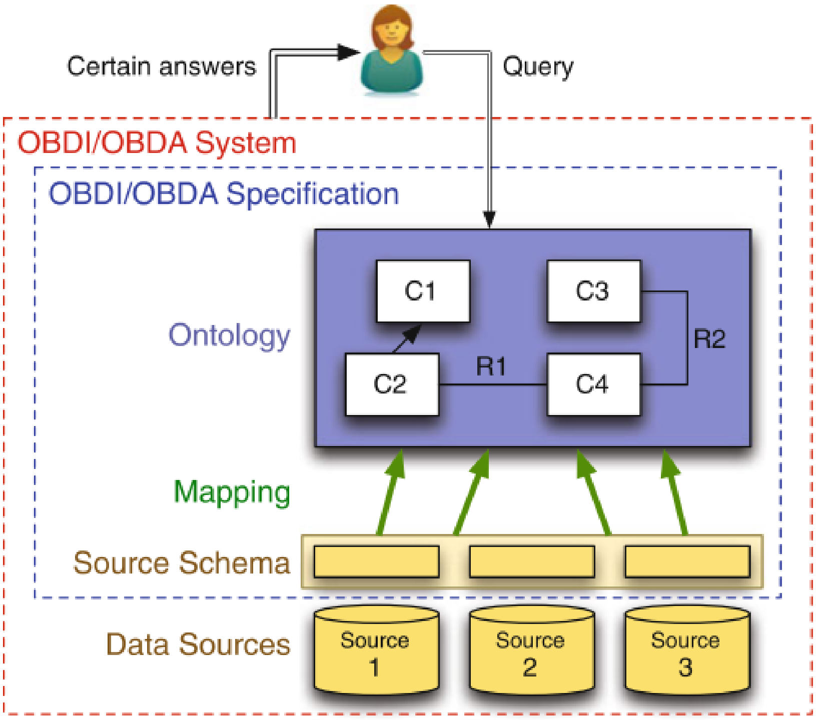


Figure 3.2: **Ontology-Based Data Access (OBDA) paradigm. Source of picture** [51].

3.1 Semantic Integration of Data

Semantic integration of data is the process of combining data from disparate sources to create a consolidated and valuable view of the information through the use of Semantic Technologies (cf. Section). The problem of integrating heterogeneous data using semantic technologies has been in the research focus for many years, in this section, we review the main approaches.

3.1.1 Ontology-Based Integration Approaches

Doan et al. [20] states that “some aspects of data integration can also be viewed as a knowledge representation problem”. Ontology-based integration approaches use an ontology as a global schema for data integration, i.e., ontologies that describes the universe of discourse. The Ontology-Based Data Access (**OBDA**) paradigm defines and uses an ontology as a core element of the data integration process [50, 51]. Calvanese et al. [51] present the three components defined by the **OBDA** paradigm: the Ontology O, the Source Schema S, and the mappings M between S and O (Figure 3.2 shows). **OBDA** paradigm is independant of its implementation that can be materialized or virtual [52], although in practice in can be both [53].

OpTop, presented by Calvanese et al. [54, 55], allows querying relational databases in an integrated manner using an ontology as the global schema. **OnTop** follows a virtual approach that means the SPARQL query is transformed into local SQL queries. **OnTop** is open source and supports the main relational databases providers. To express mappings between the source schemas and the ontology **OnTop** uses R2RML mappings. R2RML is the W3C mapping language recommendation to express mappings from relational databases to RDF [56]. R2RML has captured the attention of many researchers and several editors have been proposed [57–59] to create such mappings. Another influential work under ODBA paradigm is **Ultrawrap** [60, 61]. **Ultrawrap** takes advantage of the algorithms and optimizations already provided by database management systems (DBMS) to effectively execute SPARQL queries. **Ultrawrap** uses SQL views to encode a logical representation of RDF graphs, SPARQL queries are then translated to these views. Finally, these views query executions are automatically optimized by the DBMS providing fast response to complex SPARQL queries.

Under the OBDA paradigm, we find domain-specific applications, e.g., in the Healthcare and Clinical Data domain [62–64], where clinical trials and medical records are integrated from heterogeneous data sources. Typically, ontologies are derived from medical standards such HL7 v3 Reference Information Model[1], then a set of R2RML mappings relates the HL7 ontologies to the underlying relational databases, finally, SPARQL queries can be executed with acceptable performance. Another domain where OBDA has been applied successfully is in the Social Sciences and Humanities [65, 66], where cultural heritage data need to be integrated from heterogeneous data sources. Knoblock et al. [66] show the data integration from 14 American Art Museums, producing more than two thousand R2RML mapping rules, consolidating more than nine million RDF triples. As result, the integrated RDF Dataset can be easily queried by the different museums. Semantic data integration under the ODBA paradigm is now been studied on the Big Data scenario providing new insights to tackle the variety perspective in the Big Data scenario [67, 68]. All these domain-specific applications applied a heavy ETL integration approach.

Discussion: Despite the success of the ODBA paradigm, it has been mostly studied on integrating relational databases [69]. We argue that more research needs to be done in the scenario of heterogeneous web sources. Moreover, ETL approach is expensive, in this thesis we aim for an on-demand integration approach. Finally, interoperability conflicts such as the entity matching have not been completely addressed by ODBA that is a major challenge in the scope of this thesis. ODBA defines SPARQL the query language, we argue that SPARQL is not a suitable approach for heterogeneous web sources, which mainly provide keyword-based access mechanism, i.e., REST APIs.

3.1.2 Materialized Integration Approaches

The materialized integration approach follows an extraction-transformation-load (ETL) pipeline storing the integrated data in a single store such as Virtuoso[2], then queries and analytics can be performed on top of it. The final goal of a materialized integration approach is to produce a consolidated knowledge graph to provide smart services. Approaches towards materilized knowledge graphs include **NOUS** [70], **DeepDive** [71], and **Knowledge Vault** [72], which uses (un-,semi-)structured web sources to create a knowledge graph. **NOUS** [70] defines an end-to-end framework to create knowledge graphs for arbitrary application domains. **NOUS** combines knowledge extracted from text with curated knowledge bases, supporting the ability to answer queries where the answer is a combination of multiple data sources. Another example is **DeepDive** [71] where a full pipeline is proposed to build a knowledge graph from Wikipedia articles. **DeepDive** pipeline automatically extracts meaningful relations from the Wikipedia articles. Although **DeepDive**'s approach is generic, in the paper the authors focus on two relations: Founder/Company and Family trees. Finally, **Knowledge Vault** [72] is an automatic method for constructing a web-scale probabilistic knowledge graph. **Knowledge**

[1] http://www.hl7.org/implement/standards/rim.cfm

[2] https://virtuoso.openlinksw.com/

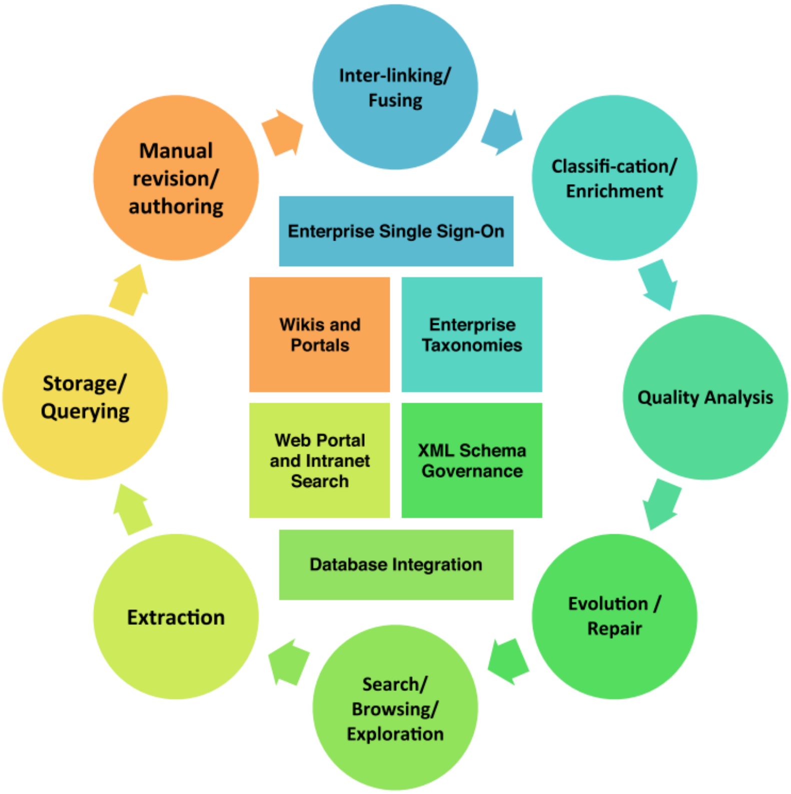


Figure 3.3: **Linked data lifecycle** for a materialized integration. Source of picture [46].

Vault combines extractions from web content with prior knowledge derived from existing knowledge bases. **Knowledge Vault** employs supervised machine learning methods for merging information from both web-crawled content and existing knowledge base facts, e.g., Freebase [73].

The traditional ETL pipeline has been adapted for the use of semantic web technologies (cf. Section 3.1). Figure 3.3 shows the life cycle proposed by Frischmuth et al. to integrate data in enterprises. A considerable amount of literature has been published on materialized integration approach using semantic technologies. Knoblock et al. [74] propose **KARMA**, a framework for integrating a variety of data sources including databases, spreadsheets, XML, JSON, and Web APIs. Using Machine Learning algorithms, **KARMA** suggests mappings from structured sources to ontologies, then using a user interface these mappings can be refined. **KARMA** has been used in several applications such as linking art data to the Linked Open Data Cloud [75] (LOD), and combating human trafficking by creating a knowledge graph of escort ads crawled from web sites [76]. Schultz et al. [77] describe the Linked Data Integration Framework (LDIF). **LIDF** is oriented to integrate RDF datasets from the Web and provides a set of independent tools to support the interlinking task. **LIDF's** tools include: (1) an expressive mapping language for translating data from various vocabularies to a consistent ontology; and (2) a Linked Data crawler component for accessing SPARQL endpoints and remote RDF dumps. **LIDF** tackles the problem of identity resolution by defining linking rules using the SILK tool [78]. Based on the defined rules, SILK identifies `owl:sameAs` links among entities of two datasets.

ODCleanStore [79] and UnifiedViews [80, 81] are another ETL framework examples for integrating RDF data. **ODCleanStore** relies on SILK to perform instance matching and provides a custom data fusion modules to merge the data of the discovered matches. Based on **ODCleanStore**, UnifiedViews supports a wide range of processing tasks including instance linking and data fusion—the LD-FusionTool [82] is responsible for data fusion. **MatWare** [83] is a tool to construct domain-specific semantic warehouses. **MatWare** focuses on connectivity assessment, provenance, and freshness of the data and it has been used to create an operational semantic warehouse in the marine domain. Finally, approaches and tools have been proposed to perform data fusion, i.e., merge data of two entities at integration time. The most remarkable examples are **Sieve** tool [84] and internal modules of the **ODCleanStore** framework.

Discussion: Although the aforementioned approaches effectively integrate heterogenous data, they require significant manual effort to configure their integration pipelines. In contrast, the purpose of this thesis is to define a universal integration pipeline that requires only a small number of high-level parameters while leaving room for tweaks and adjustments. Moreover, previous work has focused mostly on the problem of heterogeneity of data sources, while in the scope of this thesis, we focus on the problem of integrating semantically equivalent entities. In comparison, the novelty of the approaches proposed in this thesis resides in a non-materialized knowledge graph creation and profound use of Web APIs that provide access to data in web sources. Non-materialization supports efficient on-demand knowledge delivery. Further, we investigate a more suitable data integration unit that fits better the local view of the web sources data, i.e., we do not have access to the whole dataset but pieces of entity data. Thus, we research the suitability of RDF molecules to enclose the information delivered by web sources [85], we expect a more meaningful and flexible integration than ETL integration approaches.

3.1.3 Virtual Integration Approaches

Other efforts to integrate data from heterogeneous sources with semantic technologies are the virtualized integration approaches. In the virtualized approach the data remains in their local source but an intermediated layer provides an integrated access to the data sources. OpTop [54, 55] and Ultrawrap [60, 61] follow in this category, they virtualized an integrated access to heterogeneous databases, translating SPARQL queries into local SQL queries. Much of the literature on virtual integration approaches are concerned with Federated SPARQL query engines. **ANAPSID**, presented by Acosta et al. [86], is an adaptive federated query processing engine for SPARQL endpoints. **ANAPSID** is able to adapt the query execution schedules to the data availability and run-time conditions of the SPARQL endpoints. The integration step is performed by **ANAPSID** boolean operators: adaptive group-join *agjoin*, and adaptive dependent-join *adjoin*. **FedX** [87] is another example of federated SPARQL engine that enables efficient SPARQL query on heterogeneous SPARQL endpoints. **FedX** proposes join and grouping techniques to minimize the number of remote requests. First, the source selection is performed using a cache containing metadata obtained using ASK SPARQL queries from the endpoints. Then, **FedX** utilizes the *bound join* technique that uses one subquery to evaluate the input sequences producing the final result.

The federated query engine **SPLENDID** [88] optimizes SPARQL query plans using statistical data obtained from voiD descriptions [89]. **SPLENDID** provides *hash joins* and *bind joins* to optimize the performance in the query execution strategies. In **SPLENDID**, the *hash join* arguments are processed in parallel from the data sources, while in the *bind join* a variable must be bound for the succeeding join operation. Saleem et al. in [90] propose **TopFed**, a federated query engine that allows a virtual integration of multiple SPARQL endpoints. **TopFed** was

Name	Approach	Mapping-based	Schema Matching	Entity Matching	Data Fusion
OnTop [54, 55]	Virtual	YES	NO	NO	NO
UltraWrap [60, 61]	Virtual	YES	NO	NO	NO
KARMA [74]	Materialized	YES	YES	NO	NO
LIDF [77]	Materialized	NO	YES	NO	NO
ODCleanstore [79]	Materialized	NO	NO	YES	YES
MatWare [83]	Materialized	NO	NO	NO	NO
ANAPSID [86]	Virtual	NO	NO	NO	YES
FedX [87]	Virtual	NO	NO	NO	YES
SPLENDID [88]	Virtual	NO	NO	NO	YES
TopFed [90]	Virtual	NO	NO	NO	YES

Table 3.1: **Semantic Integration of Data**. Comparison of the different approaches. This is an modified version of the comparison presented by Vidal et al. [6].

evaluated with the Cancer Genome Atlas[3] (TCGA) catalog for genetic mutations responsible for cancer using genome analysis techniques. **TopFed** implements a source selection algorithm that on average selects less than half sources compared to FedX (mantaining 100 percent of recall). The authors report that, thanks to the source selection algorithm, on average **TopFed**'s query processing time is one third in comparison to state-of-the-art approaches.

Discussion: To properly apply a virtual integration using Federated SPARQL engines datasets need to be completely transformed into RDF and SPARQL endpoints need to be provided. In contrast to the Federated SPARQL approaches, in this thesis we work with Web APIs that provides a local view of the data in form of JSON objects. Moreover, we need to solve the problem of joining pieces of data from the same entity by determining relatedness, and not simply applying join operators. Thus, even if two pieces of data differ syntactically, i.e., they have non-matching URIs, they will be joined if they are identified as *semantically equivalent*.

3.2 Similarity Metrics

Entity matching is the problem of determining structured data items that describe the same real-world entity. A similarity metric tackles the problem of entity matching by comparing entities and producing a similarity value. Over the years, many research has been conducted and several approaches have been presented to measure the similarity among entities. In the context of this thesis, we review similarity metrics for RDF entities. In this section, we describe the related similarity metrics divided into three categories, i.e., rule-based, semantic-based and learning-based approaches.

3.2.1 Rule-Based Similarity Metrics

We begin by covering similarity metrics that employ handcrafted matching rules. Isele et al. propose **SILK** [78] a tool to specify matching rules that will be used to determine the similarity among entities producing `owl:sameAs` links. **SILK** also supports supervised learning, i.e., based on a dataset analysis, matching rules are automatically suggested. One of the main characteristics of **SILK** is that it offers several string similarity metrics such as Jaro distance [91] and its extension Jaro-Winkler [92]. Additionally, **SILK** matching rules can be executed on-demand through REST API requests. Ngonga et al. present **LIMES** [93] a framework to discover links among entities on the Web of Data. **LIMES** presents two novel algorithms, i.e., computation of

[3] https://cancergenome.nih.gov/

exemplars and matching-based on exemplars. These algorithms employ pessimistic estimations of distances [94] to reduce the number of comparisons necessary to complete a matching task. **LIMES** allows different approximation techniques for estimating the similarities between RDF instances. A configuration file is provided to define the properties and restrictions to measure the similarity. Similarly to SILK, **LIMES** returns `owl:sameAs` links and provide a confidence score of the matching algorithm.

Discussion: The codification of matching rules requires a deep knowledge on the domain—universe of discourse. Therefore, rule-based similarity metrics are expert dependent, i.e., experts on the field usually define the matching rules. Moreover, these approaches are hard to maintain, prone to error, and they are not flexible enough, i.e., once the rules are defined they do not adapt to any context of the data.

3.2.2 Semantic-Based Similarity Metrics

Semantic-based approaches perform an analysis at the semantic level of the entity data. Thus, no rules are manually defined but these approaches automatically analyze the semantics encoded in data, e.g., RDF knowledge graphs. Suchanek et al. present **PARIS** [95] a probabilistic approach to align RDF instances. **PARIS** detects `owl:sameAs` relationships by exploiting functional properties (a predicate that has only one object e.g., wasBornIn) of the RDF instances, to then calculate their similarity. **PARIS** does not require matching rules and offers a centralized solution tested with a large number of entities. **WebPie** [96], presented by Urbani et al., takes as input `owl:sameAs` relationships and computes in a parallel way their transitive and symmetric closure in order to produce inferred `owl:sameAs` relationships.

Jeh et al. [97] present **SimRank**, a domain-independent similarity metric that measures the neighborhood similarity among objects. The intuition behind **SimRank** is that two nodes are similar if their neighbors are similar. Thus, to compute the similarity, **SimRank** requires a full knowledge graph with all nodes and edges in it. So, closer neighbors contribute more than further nodes to the similarity value.

LINDA [98] is an automatic similarity metric that produces `owl:sameAs` links amidst entities on RDF graphs. **LINDA** produces the similarity score based on two criteria: the similarity of the data properties, and the contextual similarity that is derived from object properties of the entities. **LINDA** provides two versions of the similarity metric, i.e., a multi-core version as well as a distributed MapReduce-based version. Additionally, experiments on big RDF datasets have been made demonstrating the efficiency and scalability of **LINDA**'s algorithms.

Paul et al. [99] propose **GBSS**, an efficient graph-based document similarity. Despite the authors present **GBSS** to compare semantically annotated documents, it can be used to compare any type of entity in an RDF graph. Similar to LINDA, **GBSS** computes the similarity based on two aspects: the hierarchy of classes similarity, and the neighborhood similarity. However, **GBSS** does not take into account the entity literal properties to compute the similarity score.

Efthymiou et al. present **MinoanER** [100] a framework that discovers `owl:sameAs` relationships on RDF datasets. To reduce the number of comparisons among entities, **MinoanER** performs clustering as a pre-procession step. **MinoanER** creates clusters based on the properties of the entities as well as the metadata. Then, the similarity is applied just among the elements of the same cluster. **MinoanER** analysis the neighborhood similarity to decide whether to produce or not the `owl:sameAs` link.

Discussion: Although handcrafted rules are not necessary with the semantic-based similarity metrics, they require a complete view of the dataset. To work with the limited entity data that the web sources provide, an adaptation and evaluation are required. Moreover, the semantic-based approaches require a tuning process and its performance depends on the quality of the data. Thus, an empirical evaluation is required to measure their applicability on the knowledge retrieval and integration context from web sources.

3.2.3 Embedding-Based Similarity Metrics

With the hype of Artificial Intelligence (AI) novel approaches have been proposed to automatically create a vectorial representation of RDF entities, i.e., Graph Embeddings. Although embeddings are not directly a similarity metric, by using any vector distance metric, e.g., euclidean distance[4], they can be used to measure the similarity between entities. In this section, we provide a review of the most relevant approaches for creating embeddings for RDF entities.

Griver et al. [101] present **Node2Vec** (the latest method of the *everything-2-vec* saga). **Node2Vec** tackles the problem of producing a vector representation of graph nodes. The main contribution and uniqueness of **Node2Vec**, compared with similar techniques, is the flexible notion it gives to the meaning of neighborhood. It is based on the idea that nodes, and their connectivity patterns in the network, can be described based on two factors: First, on the communities to which they belong, i.e., homophily or essentially the set of their 1-hop neighbors, and second, on the role the nodes play in the network, i,e., structural equivalence or the type of node they are, e.g., border node, internal node, etc. Therefore, a node could have multiple neighborhoods, and it can only be considered k of these neighborhoods, the problem turns into a best-sampling-method problem. **Node2Vec** focuses on two prediction tasks: multi-label classification of nodes, where the objective is to classify new unknown nodes into one of the known classes; and link prediction with the objective of predicting if a link i.e., relation, should be established (or re-established in case of incomplete datasets) between nodes. Based on *Breadth-first Sampling (BFS)* and *Depth-first Sampling (DFS)*, **Node2Vec** proposes a sampling approach that uses *Random Walks*. It consists of exploring the connectivity patterns based on both BFS and DFS manners, interpolating between both approaches based on a bias term.

TransE, presented by Bordes et al.[102, 103], is another relevant approach to produce vector representations of entities in knowledge graphs. In **TransE**, a neural network acts as a bridge between the entities in the original graph and their feature representation, e.g., a vector of 100 dimensions. **TransE** considers only relations among entities, that means subjects and objects act as operators. The fundamental characteristics of **TransE** approach include *flexibility* and *domain independence*, i.e., it should work and be easily adaptable for most of the available knowledge graphs Additionally, the vectors produced by **TransE** are compact, each entity is assigned one low-dimensional vector in the feature space and only one matrix to each relation. In **TransE**, the relations are normal embeddings with the special characteristic that they are not normalized after each iteration of training, as for subjects and objects.

Paulheim et al. [104] present **RDF2Vec**, an RDF embeddings generation approach that adapts the word embeddings approach Word2Vec [9] for entities in an RDF knowledge graph. As noted by Paulheim: "**RDF2Vec** uses language modeling approaches for unsupervised feature extraction from sequences of words, and adapts them to RDF graphs". The idea behind **RDF2Vec** is quite simple, using Weisfeiler-Lehman Graph Kernels [105] and graph walks, **RDF2Vec** traverse the RDF graph and produces a text description of a sequence of entities. Then, Word2Vec is applied to this sequence to get embeddings for the entities in the

[4] https://en.wikipedia.org/wiki/Euclidean_distance

Name	Approach	Configuration	Output Type	Evaluated in task
SILK [54, 55]	Rule-based	Manual	same:As	Entity linking
LIMES [60, 61]	Rule-based	Manual	same:As	Entity linking
PARIS [95]	Semantic-based	Automatic	score	Entity linking
Jeh et al. [77]	Semantic-based	Automatic	score	Entity linking
LINDA [79]	Semantic-based	Automatic	score	Entity linking
GBSS [83]	Semantic-based	Automatic	score	Document similarity
MinoanER [86]	Semantic-based	Automatic	score	Entity linking and clustering
Node2Vec [101]	Embedding-based	Learned	vector	Prediction tasks
TransE [102]	Embedding-based	Learned	vector	Knowledge completion
RDF2Vec [104]	Embedding-based	Learned	vector	Classification and regression
Cochez et al. [106]	Embedding-based	Learned	vector	Classification and regression

Table 3.2: **Similarity Metrics and Graph Embeddings aproaches**. A summary of the different similarity metrics and methods to create embeddings from RDF Knowledge Graphs.

graph. The embeddings produced by **RDF2Vec** were evaluated in two simple machine learning tasks: classification, mark molecules as mutagenic or no-mutagenic; and regression, predict the lithogenesis property of rock units.

Cochez et al. [106] present an approach to generate embeddings for RDF entities based on a global-context in a given RDF graph. While RDF2Vec relies on local sequences generated from RDF graph nodes and then generate the embeddings by using Word2Vec [9], Cochez presents an approach that utilizes the global context of the graph. Inspired by GloVe [107] method—to create the embeddings for words, Cochez approach first creates a global co-occurrence matrix of entities from a given RDF graph. Then, the minimize the cost function defined in Glove is use to create the RDF entity's embeddings. Cochez's approach was evaluated with the same machine learning tasks than RDF2Vec showing competitive results, i.e, classification, and regression.

Discussion: Although RDF entity vector space embeddings have been shown to perform well in data mining and machine learning tasks -[106], these approaches have not been really applied to the data integration scenario. In addition, all these approaches required long training period time and a large set of training data. Moreover, the quality depends on the quality of the training data, so an empirical evaluation is required to test its applicability in the context of knowledge retrieval and integration from web sources.

3.3 Knowledge Retrieval and Exploration on the Web

The problem of searching for information on the Web has evolved during the last decades. As a result, the Information Retrieval (IR) is the most mature research field in this area. As Herzig [1] states: "Nowadays, search on the Web goes beyond the retrieval of textual Web sites and increasingly takes advantage of the growing amount of structured data". Companies, like Google, have recognized this and now provide a semantic entity search as part of its engine [10]. In the scope of this thesis, we review three main trends in terms of entity retrieval and exploration in the Web: (1) **hybrid search engines** that combine structured, semistructured, and unstructured data to produce the results; (2) **domain-specific search engines** that are created with a specific domain knowledge on the mind; and (3) **faceted search engines** for RDF, the data model selected on this thesis. These approaches are explained in the following sub-sections:

3.3.1 Hybrid Search Engines

Several approaches have been proposed to combine search results from unstructured data, e.g., text documents, with structured data, e.g., RDF. For example, Usbeck et al. [5] present a Hybrid Question Answering Framework (**HAWK**) combining entity search over linked data and textual data from the Web. The search input is a question expressed in natural language, which passes through an eight-step pipeline. **HAWK**'s pipeline is quite complex since it pursues a question answering on natural language queries. So the keyword search APIs, provided by web sources, are not considered in its pipeline. Bhagdev et al. [4] propose a hybrid search architecture that aims the combination of concepts search and keyword search on documents and their metadata. They propose a keyword search engine on documents and combined with the search on documents metadata. The architecture proposed by Bhagdev et al. requires an index of the documents. Moreover, they focus only on documents, leaving the gap of searching for a more generic approach that works on the abstract concept of an entity.

Montoya et al. [108, 109] proposes **SemLAV**, a hybrid search engine to query Deep Web and Web of Data sources. **SemLAV** executes SPARQL queries against the Deep Web and Linked Open Data data sources by using a mediator-wrapper architecture approach (cf. Section 2.3). The SPARQL queries are expressed using a mediator schema vocabulary, then **SemLAV** selects relevant data sources and ranks them. **SemLAV** ranking strategy delivers results quickly based only on view definitions, thus no statistics on data sources are required.

Herzig et al. [1] presents a entity search engine that consolidates entities from heterogeneous data sources on-the-fly. The authors propose a language-model based approach to represent the entities coming from web sources and to compute the similarity among the entities. Herzig et al. use the concept of a threshold value to perform an entity consolidation step, i.e., only entities which similarity value pass the threshold are merged. Herzig's approach is the most similar work to the one presented on this thesis. Nevertheless, the representation of entities and the similarity metric is static, they are strongly correlated and cannot be replaced for a more suitable approach according to the domain-application context.

Discussion: The approaches presented in this section are first attempts have been made to provide a unified search across unstructured (Web) and structured (RDF) sources dubbed *hybrid search*. We argue that a much more universal approach for a federated hybrid search encompassing not only unstructured and RDF sources is required to address application scenarios described in this thesis. In particular, various degrees of structure (unstructured, semi-structured and structured), various data models and data topologies (distributed, federated and integrated data sources) have to be supported. In this thesis, we aim to provide an approach and its prototypical implementation for a federated semantic search. Our approach is capable to retrieve and integrate the knowledge about entities spread on web sources that, albeit described differently, correspond to the same real-world entity.

3.3.2 Domain Specific Search Engines

In the specific application domain of law enforcement, organizations are demanding more intelligent software to support their work. Therefore, both the academia and the industry are making efforts to build innovative crime analysis software. The **DIG** system [76] builds a knowledge graph to combat human trafficking by crawling websites with escort ads. The **DIG** system provides an easy to use faceted browsing interface to query and explore the knowledge graph. **Huber** [110] presents a crime investigation tool focused on social networks. Huber's tool approach harvest data from social networking websites, e.g., user data, private messages, photos, etc. To produce a consolidated profile information about people and organizations. Then, these profiles can be explored via a user interface. **Maltego**[5] is an open source forensics application. **Maltego** offers information mining as well as visualization tools to determine the relationships between entities such as people, companies, or websites. Finally, **Poderopedia**[6] is an initiative to promote transparency of power control in South America. **Poderopedia's** search engine relies on a knowledge graph containing people and the power they have on the continent. Journalists contribute by adding entities and relations to the knowledge graph. The degree of power a person has is determined by its relations with other organizations and people.

In the Biomedical domain, several search engines have been proposed to integrate heterogeneous web sources. Hu et al. [111] present **BioSearch** a search engine that uses ontologies to execute federated queries over SPARQL endpoints. **BioSearch** utilizes a virtual integration approach to provide a unified view of heterogenous RDF datasets in the biomedical domain. **BioSearch** applies an ontology matching approach to solve the heterogeneity of schema problem. Therefore, mappings are created to convert local RDF entities to a global schema, i.e., the SIO ontology [112].

Discussion: Despite these search engines solved domain-specific scenarios, we argue that they are still expensive to maintain. Mainly because they require a consolidated knowledge graph. Additionally, privacy issues need to be taken into consideration, especially in the law enforcement domain (one of the main use cases in this thesis), where data protection and privacy laws, e.g., the General Data Protection Regulation, must be followed. We see the need for a more suitable set of tools to quickly adapt to dynamicity of the web sources. In contrast to these search engines, we aim an approach to building knowledge graph on-demand, i.e., when a query is entered, the results are built by integrating results collected from search APIs.

3.3.3 Faceted Search Engines

Entity search is one of the main use cases in the scope of this thesis. In this section, we review faceted search engines over RDF, the data model selected in this work. Arenas et al. [113, 114] introduced **SemFacet** a faceted search approach on RDF knowledge graphs. **SemFacet** automatically generates facet names and values from metadata provided in RDFS and OWL. Besides the use of explicit knowledge encoded RDF entities, **SemFacet** creates more facets from the implicit knowledge by using a reasoning component on the RDF knowledge graph. **SemFacet** applies inference algorithms to derive new facts about the entities, which are then used as new facets. **SemFacet** is based on a strong theoretical framework that accounts for both RDF data and OWL2 axioms [115].

Stadler et al. [116] presented **Facete**, a spatial data search engine that query SPARQL endpoints to produce a faceted view of the entities. Thus, a difference of SemFacet, **Facete** allows exploring multiple RDF knowledge graphs at the same time. **Facete** automatically generates the facets and provides a map view of the spatial data for the users to explore. However, **Facete** focus on the exploration of spatial data, i.e., its application to another type of data will require the development of extensions.

Ferré [117] presents **Sparklis** faceted search, the goal of **Sparklis** is to enable non-technical users to explore RDF entities. Using SPARQL endpoints, **Sparklis** allows to answer complex questions based on facets suggested by the application. To do so, **Sparklis** combines the

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

[6] http://www.poderopedia.org/

expressivity of SPARQL query language and the usability of faceted search, i.e., the facets execute SPARQL queries to retrieve answers to complex questions.

Finally, Khalili et al. [118] presented **LD-R**, a component-based framework to quickly build user interfaces for RDF knowledge graphs. Although **LD-R** focuses on providing a development framework, one of its main out-of-the-box configurations is the faceted browsing user interface named **FERASAT** [119]. **FERASAT** is a novel faceted browsing environment, it supports a set of serendipity-fostering design patterns in the facets—serendipity allows the discovery of valuable facts not initially sought for.

Discussion: Faceted Search Engines on RDF allows to search and explore knowledge about entities in an RDF knowledge graph. Faceted browsing is the defacto entity exploration approach in the Semantic Web community, but these approaches work under the assumption of having the access one consolidated RDF graph. Thus, we argue the exploration of entities coming from a federation of sources, which is the scenario in this thesis, is still underexplored. In this thesis, we propose a reactive user interface to explore entities that come from multiple sources, i.e., a user interface that reacts and adapts itself properly to the heterogeneity of data and semantics encoded in the entities. Stolz and Hepp [120] conducted an evaluation of the appropriateness of a reactive faceted search user interface paradigm for e-commerce on the Web of Data, i.e., the user interface elements changes according to the semantics of the data. In their work, they present preliminary evidence of the applicability of an adaptive user interface for faceted search. In this thesis, we explore the applicability of this approach in the scenario of knowledge retrieval and integration from web sources.

CHAPTER 4

Semantic Based Approaches for Synthetizing Equivalent Entities

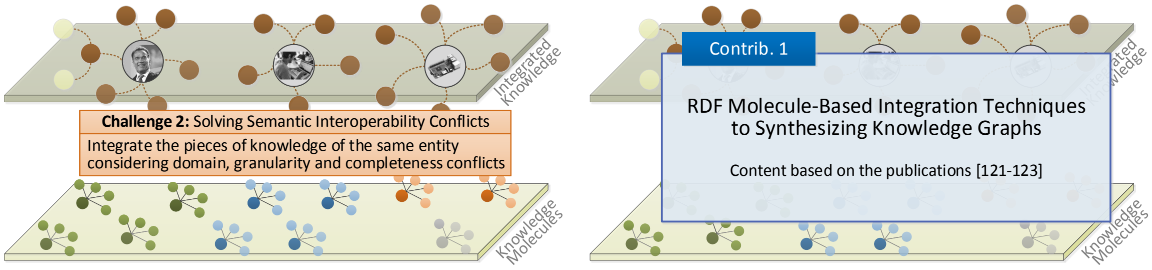
This chapter is dedicated to solve one of the core challenges of this thesis, i.e., to identify and integrate knowledge of semantic equivalence entities. The content of this chapter is based on the publications [121–123]. The nature of the Web allows for numerous descriptions of the same entity, generating data interoperability conflicts (cf. Section 2.3.5). Integrating data from web sources requires the effective identification and resolution of these interoperability conflicts. Figure 4.1(a) shows the main problems to produce an integrated knowledge from pieces spread over web sources. The results of this chapter provide an answer to the following research question:

RQ1: How can semantics encoded in RDF graphs be exploited during the process of integrating data collected from heterogeneous web sources?

To provide a unified representation of the same real-world entity, the data contained in web sources need to be semantically integrated. Therefore, we require a semantic integration approach capable of managing and exploiting the knowledge encoded in web sources to determine the relatedness of different representations of the same entity, e.g., axiomatic definition of vocabulary properties or resource equivalences. We assume at this level that the heterogeneity of representation problem is solved (cf. Section 1.2.1), and we work with the RDF data model.

First, in Section 4.1, we present a motivating example illustrating the problem of integrating semantically equivalent RDF entities. To address research question **RQ1**, we devise MINTE, a semantic integration technique able to utilize semantics encoded in vocabularies in order to fuse semantically equivalent RDF entities in a single pipeline—what we call semantic integration. Next, Section 4.2 describes our approach including a formal problem statement and the main steps MINTE performs. MINTE implements a two-fold approach for both determining the relatedness of two RDF entities and merging them. Section 4.2.2 details our semantic disambiguation and integration technique, and the data fusion strategy and policies for merging equivalent RDF molecules. Then, the main two properties of MINTE, i.e., **high adaptability** and **low complexity** of the integration approach are presented in Section 4.3.

A comprehensive evaluation of the MINTE approach and analysis of the obtained results is presented in Section 4.4. Observed results suggest that MINTE is able to accurately integrate *semantically equivalent* RDF graphs. Further, MINTE behavior is empowered by semantic similarity measures, ontologies, and fusion policies that consider not only textual data properties


Integrated Knowledge
Challenge 2: Solving Semantic Interoperability Conflicts
Integrate the pieces of knowledge of the same entity considering domain, granularity and completeness conflicts
Knowledge Molecules
Contrib. 1
RDF Molecule-Based Integration Techniques to Synthesizing Knowledge Graphs
Content based on the publications [121-123]
Integrated Knowledge
Knowledge Molecules

(a) Problems tackled in this chapter

(b) Contributions described in this chapter

Figure 4.1: **Challenges and Contributions:** This chapter focuses on the problem of integrating knowledge of semantically equivalent entities from different web sources, and propose an RDF molecule-based integration approach to solve this problem.

as current approaches do, but also logical axioms encoded in the graphs to tackle relations among objects and properties. Finally, Section 4.5 presents the closing remarks of this chapter. We summarize the contributions of this chapter as follows:

- A novel semantic integration approach named MINTE, which is based on the concept of RDF molecules of knowledge. MINTE clearly defines the use of semantic technologies as building blocks and configuration parameters in the integration process.

- A novel method for matching and merging semantically equivalent RDF entities. Semantics encoded in RDFS and OWL are exploited during the integration process.

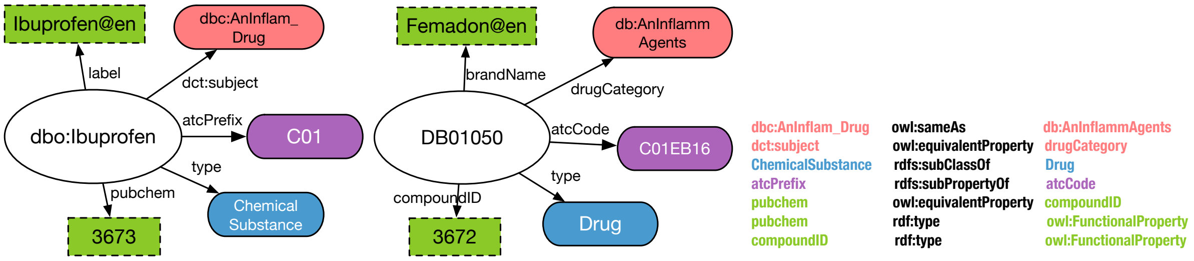
- An empirical evaluation to assess the effectiveness of MINTE for the integration of RDF graphs. Experiments are executed on DBpedia, Wikidata, and Drugbank. Different types of heterogeneity at schema, property, and value levels are considered in the study.

4.1 Integrating Semantic Equivalent Entities

The original vision of the Web put a strong emphasis on the distributed and federated nature. the Semantic Web follows the same vision. While there have been some efforts to provide a unified view of the entities contained on web sources, such as federated SPARQL queries [86–88], semantic search and (meta-)data registries, we still deem that there is an imbalance and a large part of the data integration technologies are mimicking traditional data management techniques. The knowledge about entities is spread over different web sources on the internet or even in the intranets of organizations. For example, information about chemical components and drugs is published by different data providers, e.g., *DrugBank*[1] or *Kegg*[2]. Similarly, data about people can be found in social networks, customer relationship management (CRM), or human-resource (HR) systems. Further, product information is available in e-commerce web sites, product life-cycle management (PLM) systems or open product data repositories. It is not realistic to expect that all these data sources will publish their data using the same unique identifiers and unified vocabularies.

[1] https://www.drugbank.ca/

[2] http://www.genome.jp/kegg/


Ibuprofen@en
label
dbc:AnInflam_ Drug
dct:subject
dbo:Ibuprofen
atcPrefix
C01
type
Chemical Substance
pubchem
3673
Femadon@en
brandName
db:AnInflamm Agents
drugCategory
DB01050
atcCode
C01EB16
type
Drug
compoundID
3672
dbc:AnInflam_Drug owl:sameAs db:AnInflammAgents
dct:subject owl:equivalentProperty drugCategory
ChemicalSubstance rdfs:subClassOf Drug
atcPrefix rdfs:subPropertyOf atcCode
pubchem owl:equivalentProperty compoundID
pubchem rdf:type owl:FunctionalProperty
compoundID rdf:type owl:FunctionalProperty

(a) DBpedia RDF Molecule (b) Drugbank RDF Molecule (c) Axioms

Figure 4.2: **Motivating Example.** The drug Ibuprofen in the DBpedia and Drugbank RDF graphs. Properties such as *name* or *case number* are shared in both RDF graphs, while properties such as *chemical formula* or *name translations* only exist in one of the graphs. The challenge is to produce an integrated RDF molecule for Ibuprofen.

The integration of semantic equivalent entities is an important task in a variety of domains but is it hard due to the semantic interoperability problems. For a motivating example, we choose the chemical domain, where numerous representations of drugs are available across various RDF graphs. All of them are valid RDF descriptions despite using different schemas or covering different properties. DBpedia contains general information about drugs, for instance, `dbr:Ibuprofen`[3] (Figure 4.2(a)) comprises common properties, e.g., `rdfs:label` in different languages, `dct:subject` categories, rich `rdf:type` annotations in terms of numerous ontologies (i.e., DBpedia ontology, YAGO, Umbel, Wikidata, DUL), and links to other language pages in DBpedia, wiki links and knowledge bases. Another well-known dataset in the chemical domain is Drugbank. Drugbank's description of Ibuprofen[4] (Figure 4.2(b)) contains detailed, domain-specific drug data i.e., chemical formula, pharmacological data, interactions, drug targets, enzymes, and transporters. Drugbank and DBpedia descriptions have few facts in common but greatly complement each other. Although the vocabularies, URIs, properties and values used to describe the drugs are different, they refer to the same real world drug.

Semantic technologies provide the basis for semantic description, interlinking and fusion of disparate web sources. Existing approaches often separate the linking and fusion steps, which we together subsume under the concept of semantic integration. Linking approaches implemented in tools such as *Silk* [78] and *LIMES* [93], for example, allow for discovering links between RDF resources by exploiting the similarity of literals of their datatype properties. Thus, entities representing drugs with similar names can be linked. Subsequently, data fusion frameworks such as *Sieve* [84] and *ODCleanStore* [79] implement methods for semi-automatically merging equivalent RDF entities. However, we believe that both data linking and fusion approaches do not sufficiently exploit the semantics encoded in the vocabularies used to describe heterogeneous data, e.g., functional and inverse properties, sub-classes, or sub-properties. Consequently, RDF entities that are represented using syntactically different properties or resources, but that are semantically equivalent with respect to a vocabulary semantics, cannot be linked or integrated.

[3] `http://dbpedia.org/page/Ibuprofen`

[4] `https://www.drugbank.ca/drugs/DB01050`

4.2 The MINTE approach

4.2.1 Problem Definition

At the conceptual level, we model the pieces of knowledge spread on web sources as RDF molecule for knowledge integration. Thus, the problem of integrating semantic equivalent entities from different web sources can be defined as follows:

Definition 1 (RDF Molecule) *Given an RDF graph G, we call a subgraph M of G an RDF molecule iff the RDF triples of $M = \{t_1, \ldots, t_n\}$ share the same subject, i.e., $\forall\ i, j \in \{1, \ldots, n\}$ $(subject(t_i) = subject(t_j))$. An RDF molecule can be represented as a tuple $\mathcal{M} = (R, T)$, where R corresponds to the URI (or blank node ID) of the molecule's subject, and T is a set of pairs $p = (prop, val)$ such that the triple $(R, prop, val)$ belongs to M. Property values are free of blank nodes, i.e., let I be a set of IRIs and L a set of literals, then $val \in I \cup L$.*

We call R and T the *head* and the *tail* of the RDF molecule $\mathcal{M}$, respectively. For example, an RDF molecule of the drug *Ibuprofen* is (`dbr:Ibuprofen`, {(`rdf:type`, `ChemicalSubstance`), (`dbo:actPrefix`, `"C01"`), (`pubchem`, `3673`)})[5]. Further, an RDF graph G is defined in terms of RDF molecules as follows:

$$\phi(G) = \{\mathcal{M} = (R, T) | t = (R, prop, val) \in G \wedge (prop, val) \in T\}$$

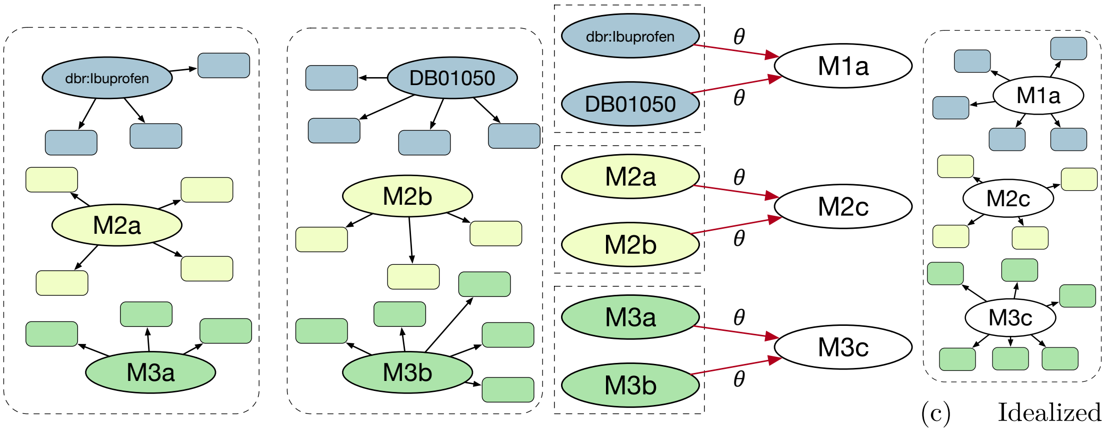
Definition 2 (Equivalent RDF molecules) *Let $\phi(G)$ and $\phi(D)$ be two sets of RDF molecules. Let F be an idealized set of integrated RDF molecules across all sets of RDF molecules. Let θ be a homomorphism such that $\theta : \phi(G) \cup \phi(D) \to F$. Let $\mathcal{M}_G$ and $\mathcal{M}_D$ be RDF molecules from $\phi(G)$ and $\phi(D)$, respectively. $\mathcal{M}_G$ and $\mathcal{M}_D$ are semantically equivalent if and only if there is an RDF molecule $\mathcal{M}_F$ from F, such that $\theta(\mathcal{M}_D) = \theta(\mathcal{M}_G) = \mathcal{M}_{\mathcal{F}}$.*

Given two RDF graphs G and D, an entity e can be represented by two different RDF molecules $\mathcal{M}_G$ and $\mathcal{M}_D$ in $\phi(G)$ and $\phi(D)$, i.e., $\mathcal{M}_G$ and $\mathcal{M}_D$ corresponding to semantically equivalent RDF molecules. In this work, we tackle the problem of matching and merging semantically equivalent RDF molecules from RDF graphs. This problem is defined as follows: Given $\phi(G)$ and $\phi(D)$ composed of RDF molecules, and an idealized set F of integrated RDF molecules from $\phi(G)$ and $\phi(D)$ and free of semantically equivalent RDF molecules, i.e., there is only one RDF molecule in F that corresponds to the integration of $\mathcal{M}_G$ and $\mathcal{M}_D$.

The *problem of semantically integrating* $\phi(G)$ and $\phi(D)$ consists of building a homomorphism $\theta : \phi(G) \cup \phi(D) \to F$, such that if RDF molecules $\mathcal{M}_G$ and $\mathcal{M}_D$ represent the same entity e, then $\theta(\mathcal{M}_G) = \theta(\mathcal{M}_D)$, otherwise, $\theta(\mathcal{M}_G) = \theta(\mathcal{M}_D)$.

Consider the RDF molecules presented in Figure 4.3 to illustrate an instance of the *problem of semantically integrating* RDF graphs. RDF molecules in Figure 4.3(a) belong to two different datasets but they are semantically equivalent because both represent the same entity, i.e., the Ibuprofen drug. On the other hand, RDF molecules in Figure 4.3(c) comprise an idealized set of RDF molecules that integrates semantically equivalent molecules from two graphs in Figure 4.3(a). A solution of the *problem of semantically integrating* RDF graphs is to identify the homomorphism θ that maps RDF molecules (e.g., Ibuprofen and DB01050) into integrated RDF molecules. Figure 4.3(b) illustrates the homomorphism to map source graphs in Figure 4.3(a) to an idealized graph in Figure 4.3(c).

[5] We use standard prefixes according to [6]



(a) Two RDF Graphs

(b) Homomorphism θ

(c) Idealized RDF Graph

Figure 4.3: **The problem of semantically integrating RDF graphs.** The two RDF graphs in 4.3(a) contain semantically equivalent RDF molecules. The problem consists of building a homomorphism θ to an idealized integrated RDF graph such as that in 4.3(c). Such a homomorphism θ to map equivalent entities, e.g., `dbr:Ibuprofen` and `DB01050`, to an integrated entity is presented in 4.3(b).

However, real-world cases impose several restrictions on the problem of building a homomorphism. Firstly, only few RDF graphs provide their entities with established links to the semantically equivalent entities in other graphs. Generally, such links must be discovered beforehand. Secondly, an ideal graph is hardly ever available for the entire diversity of RDF graphs available in Linked Open Data cloud. Therefore, the initial problem has to be *approximated* taking into account the real-world settings. The approximation includes two steps: *identification* of semantically equivalent entities in arbitrary RDF graphs and *fusion* of the matches found into an idealized unified RDF graph. For instance, during the identification task it has to be decided whether `dbo:Ibuprofen` (Figure 4.2(a)) and `DB01050` (Figure 4.2(b)) are equivalent based on their properties and general axioms in Figure 4.2(c). In turn, the fusion task aims at generating a unified representation of those two molecules if they are marked as equivalent. Below we describe how our approximation approach tackles the identification and fusion steps to build a homomorphism between arbitrary RDF graphs.

4.2.2 Proposed Solution

We propose MINTE, an integration framework able to identify and merge semantically equivalent RDF graphs, thus providing a solution to the *problem of semantically integrating* RDF graphs. MINTE consists of two essential components. First, the identification component discovers semantically equivalent entities with the help of two sub-components, namely the Dataset Partitioner and the 1-1 Weighted Perfect Matching Calculator. Second, the Integrator component digests the output of the previous one in order to produce a semantically integrated knowledge graph. Figure 4.4 depicts the main components of the MINTE architecture. The pipeline receives two RDF graphs G and D, and additional parameters in order to produce a semantically integrated RDF graph. MINTE relies on a semantic similarity measure Sim_f and an ontology O to determine when two RDF molecules are semantically equivalent.

Semantic similarity functions employ the axioms in O together with the object properties (cf. Figure 4.5(a)) to deduce a semantic equality of such entities. Additional knowledge

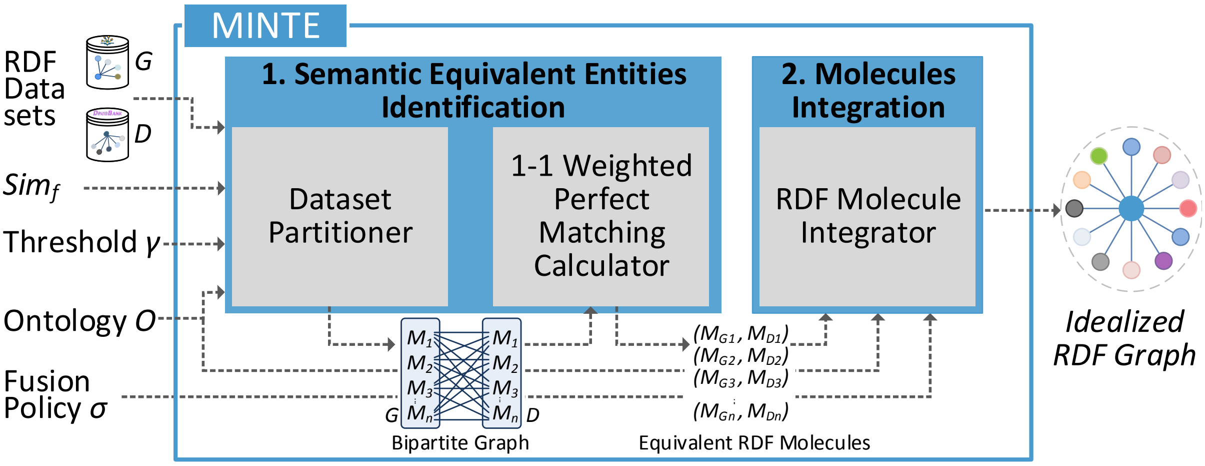
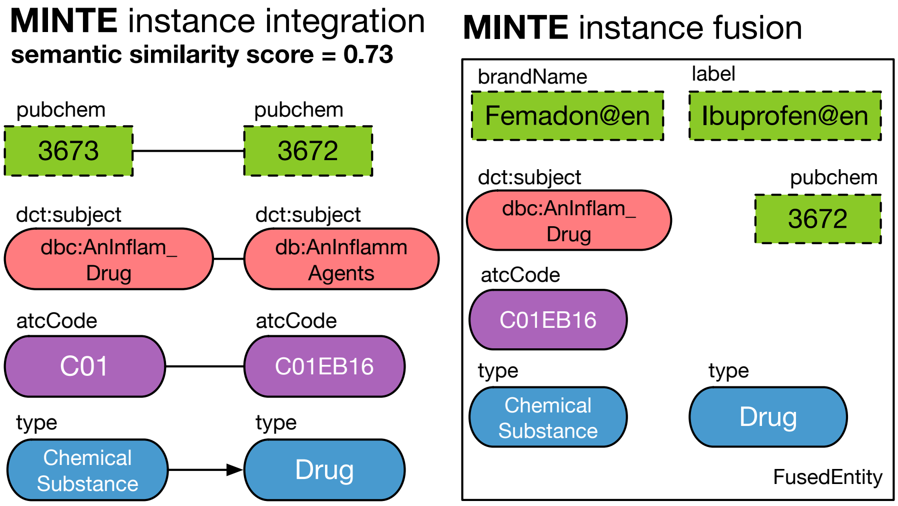


Figure 4.4: **The MINTE Architecture.** MINTE receives RDF datasets, a similarity function Sim_f, a threshold γ, an ontology O, and a fusion policy σ. The output is a semantically integrated RDF graph. A Dataset Partitioner creates a bipartite graph of RDF molecules and assigns the similarity value according to Sim_f, γ, and O. Semantically equivalent RDF molecules are related by edges in a 1-1 weighted perfect matching from the bipartite graph. Equivalent RDF molecules are integrated according to σ and mappings in O

about class hierarchy (rdfs:subClassOf), equality of objects (owl:sameAs) and properties (owl:equivalentProperty) allows to investigate deep semantic relations at the graph level instead of comparing plain literals. For instance, analyzing Figure 4.2(c) one can entail that ChemicalSubstance is a sub-class of Drug, anti-inflammatory drug is the same entity as anti-inflammatory agent, the values of atcPrefix and atcCode of the compared drugs are close to each other, and finally that "Ibuprofen" is synonymous to "Femadon". Therefore, a semantic analysis manages to discover that dbo:Ibuprofen and DB01050 are semantically equivalent.

To do so, the Dataset Partitioner compares RDF molecules in $\phi(G)$ and $\phi(D)$ based on the similarity measure Sim_f. A bipartite graph is created between G and D; edges correspond to the pair-wise comparison of the RDF molecules and are weighted with values of the similarity measure Sim_f. Once a bipartite graph is created, MINTE identifies the *semantically equivalent* RDF molecules. A 1-1 weighted perfect matching algorithm is executed in order to identify for each RDF molecule the most similar one. Thus, if two RDF molecules are connected by an edge of the 1-1 perfect matching, then they are considered *semantically equivalent*.

Finally, the RDF Molecule Integrator component resorts to fusion policies σ for integrating semantically equivalent RDF molecules and generating an integrated RDF graph. An ontology O is utilized to map properties and resources in equivalent RDF molecules, while fusion policies σ specify certain rules for how the mapped properties or values should be physically merged in order to eliminate redundancy while preserving consistency. Figure 4.5(b) illustrates how semantic fusion is capable of producing a fused entity aiming at complete and consistent facts. Between two objects linked by owl:sameAs only one, e.g., dbc:AnInflammDrug, is chosen; one atcCode property is merged into a fused entity as it contains a more general and complete value; class hierarchy is retained; pubchem value is kept as a functional property, i.e., it should have only one value; and labels with brand names complement each other.



(a) Instance integration

(b) Instance fusion

Figure 4.5: Instance and integration fusion.

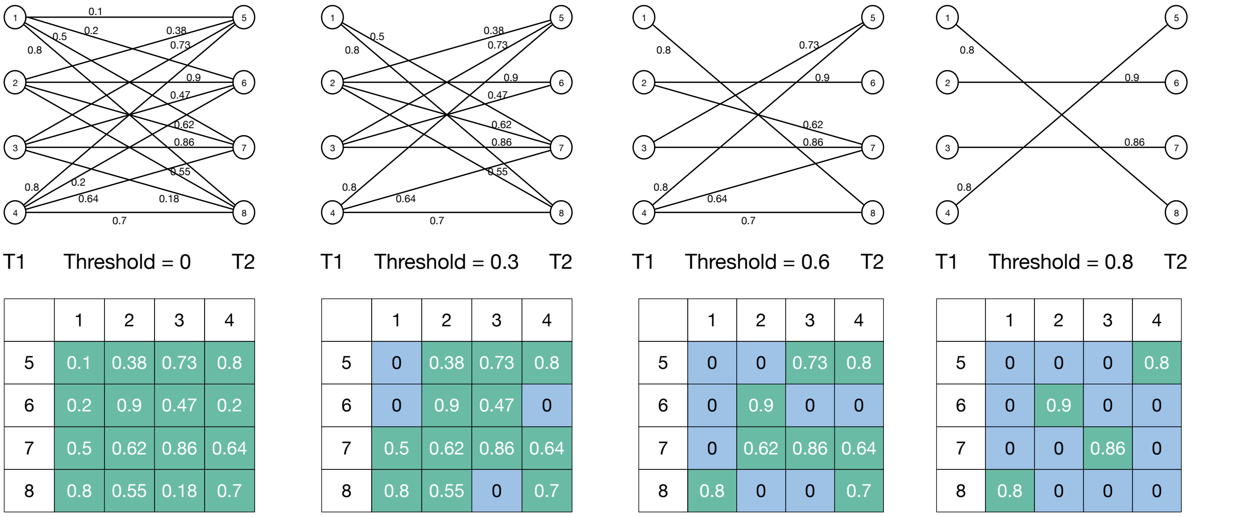
Identifiying Semantic Equivalence Entities

MINTE uses a semantic equivalent technique to decide when two RDF molecules correspond to the same entity, e.g., determining if two drugs are semantically equivalent. The process involves two stages: (a) dataset partitioning and (b) finding a perfect matching between partitions.

Dataset Partitioner. The partitioner employs a similarity measure Sim_f and ontology O to compute the relatedness between RDF molecules in $\phi(G)$ and $\phi(D)$. Addressing flexibility, MINTE allows for arbitrary, user-supplied similarity functions that leverage different algorithms to estimate the extent of correlation between RDF molecules. Supporting a variety of similarity measures including simple string similarity functions we, however, advocate semantic similarity measures that achieve better results (as we show in Section 4.4) by considering semantics encoded in RDF graphs. After computing similarity scores, the partitioner constructs a bipartite graph between the sets $\phi(G)$ and $\phi(D)$.

A threshold γ is used to discard edges of the graph whose weights are lower than γ. Figure 4.6 illustrates the impact of different threshold values on the number of edges of a bipartite graph. Edges in bipartite graphs represent relations between RDF molecules with similarity values greater or equal than a threshold. If the threshold is equal to 0, the bipartite graph is complete and the edges represent the pair-wise comparison of the RDF molecules. Contrarily, if the threshold is high, e.g., 0.8, few edges are included in the graph.

1-1 Weighted Perfect Matching. MINTE solves the problem of identifying *semantically equivalent* RDF molecules by computing a 1-1 weighted perfect matching between the sets of RDF molecules to be integrated. The input of the 1-1 weighted perfect matching component is a weighted bipartite graph, where a weight of an edge between two RDF molecules corresponds to a similarity value. The Hungarian algorithm is utilized to compute the matching. Figure 4.7(b) illustrates the result of computing a 1-1 weighted perfect matching on the bipartite graph in Figure 4.7(a). The edges between the RDF molecules in the graph in Figure 4.7(b) represent the fact that the connected RDF molecules are semantically equivalent, e.g., RDF molecules *M2a* and *M2b* are semantically equivalent. As will be shown in the results reported in 4.4, the accuracy



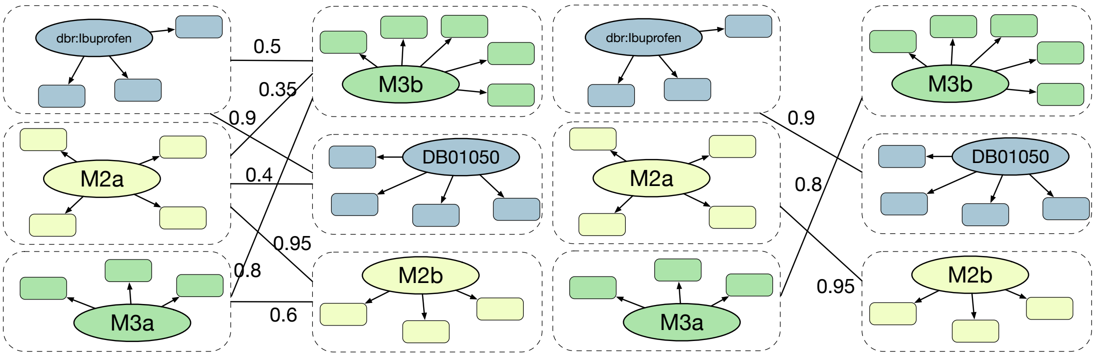
	1	2	3	4
5	0.1	0.38	0.73	0.8
6	0.2	0.9	0.47	0.2
7	0.5	0.62	0.86	0.64
8	0.8	0.55	0.18	0.7

	1	2	3	4
5	0	0.38	0.73	0.8
6	0	0.9	0.47	0
7	0.5	0.62	0.86	0.64
8	0.8	0.55	0	0.7

	1	2	3	4
5	0	0	0.73	0.8
6	0	0.9	0	0
7	0	0.62	0.86	0.64
8	0.8	0	0	0.7

	1	2	3	4
5	0	0	0	0.8
6	0	0.9	0	0
7	0	0	0.86	0
8	0.8	0	0	0

Figure 4.6: **Bipartite Graph Pruning.** Different thresholds on the values of the similarity measure and the impact on a bipartite graph between RDF molecules. A threshold equal to 0.0 does not impose any restriction on the values of similarity; thus the bipartite graph includes all the edges. High thresholds, e.g., 0.8, restrict the values of similarity, resulting in a bipartite graph comprising just a few edges.



(a) Initial Bipartite Graph, T=0.3

(b) 1-1 Weighted Perfect Matching

Figure 4.7: **1-1 Weighted Perfect Matching.** (a) A bipartite graph between RDF molecules from DBpedia and Drugbank; only the edges with similarity values equal or greater than 0.3 are included in the graph. (b) A 1-1 weighted perfect matching of the graph in (a); each RDF molecule is matched to the most similar one.

of the process of determining when two RDF molecules are semantically equivalent is impacted by the characteristics of the similarity measure Sim_f. In case a semantic similarity measure like $\mathcal{GADES}$ is utilized, MINTE is able to *precisely* match RDF molecules that correspond to the same real-world entity.

Integration Semantic Equivalence Entities

Once the semantically equivalent RDF molecules have been identified, the second component of MINTE produces an integrated knowledge graph. In order to retain completeness and consistency and, at the same time, reducing the redundancy of the data, MINTE applies a set σ of *fusion policies*, i.e., rules operating on the triple level, which are triggered by a certain combination of predicates and objects. Fusion policies resemble flexible filters tailored for specific tasks, e.g., keep all literals with different language tags or discard all except one, replace one predicate with another, or simply merge all predicate-value pairs of given molecules. Fusion policies resort to an ontology O to resolve possible conflicts and inequalities on the levels of resources, predicates, objects and literals.

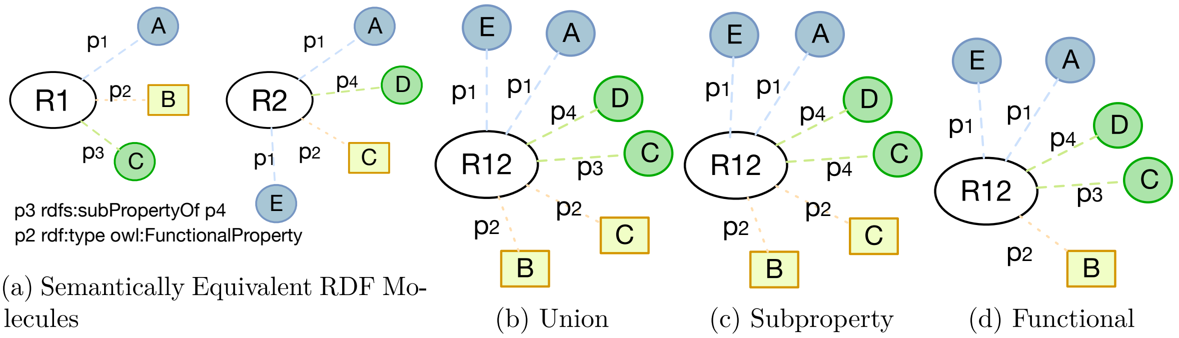
The policies that process resources, e.g., URI naming conventions when creating an integrated graph, are denoted as a subset $\sigma_r \in \sigma$. The policies that focus on properties are denoted as $\sigma_p \in \sigma$. Interacting with the ontology O, σ_p tackles RDFS and OWL property axioms, e.g., `rdfs:subPropertyOf`, `owl:equivalentProperty`, and `owl:FunctionalProperty`. Such an interaction is particularly important when the σ_p policies have to comply with sophisticated OWL restrictions on properties. That is, if a certain property can have only two values of some fixed type, σ_p has to accurately monitor the merging process to ensure semantic consistency. Lastly, the policies dedicated to objects (both entities and literals) comprise a subset $\sigma_v \in \sigma$. For literals, the σ_v policies have to implement string processing techniques, such as recognition of language tags, e.g., *@en*, *@de*, etc., to be able to identify whether two values are different or the same but with syntactic errors. For instance, `S1 rdfs:label "Ibuprofen"@en` and `S1 rdfs:label "Aktren"@de` are considered different whereas `"Ibuprofen"@en` and `"Iburpofen"@en` are evidently the same. Similar requirements can be applied to `xsd:date` and other standard datatypes. For objects of object properties, the σ_v policies are more flexible and may provide rules in case, e.g., objects of different properties are linked by `owl:sameAs`. Generally, the σ_v policies are closely connected with the σ_p policies and affect each other, allowing for enriching an integrated knowledge graph with new facts. For instance, some configuration of σ_p and σ_v may lead an OWL reasoner to deduce from `:Person :birthCity ns1:Berlin` and `:Person :birthCity ns2:Q64` that `ns1:Berlin` and `ns2:Q64` are `owl:sameAs`. MINTE defines three fusion policies, which are illustrated in Figure 4.8:

Union policy. The union policy creates a set of $(prop, val)$ pairs where duplicate pairs, i.e., pairs that are syntactically the same, appear only once. In Figure 4.8(a) the pair (p_1, A) is replicated, then it is included once in Figure 4.8(b). The rest of the pairs are added directly.

Subproperty policy. This policy tracks if a property of one RDF molecule is a sub-property (`rdfs:subPropertyOf`) of a property of another RDF molecule, i.e., $\{r_1, p_1, A\}, \{r_2, p_1, B\} + O + subPropertyOf(p_1,\ p_2) \models \{\sigma_r(r_1, r_2), p_2, \sigma_v(A, B)\}$. As a result of applying this policy, the property p_1 is replaced with a more general property p_2. The default σ_v object policy is to keep the property value of p_1 unless a custom policy is specified. In Figure 4.8(c), a property p_3 is generalized to p_4 while preserving the value C according to the ontology axiom `p3 rdfs:subPropertyOf p4` in Figure 4.8(a).

Authoritative graph policy. The policy allows for selecting one RDF graph as a prevalent source of data when integrating the following configurations of $(prop, val)$ pairs:

- The **functional property policy** keeps track of the properties annotated as funtional properties (`owl:FunctionalProperty`), i.e., such properties may have only one value. The authoritative graph policy then retains a value of a molecule from the primary graph: $\{r_1, p_1, B\}, \{r_2, p_1, C\} + O + functional(p_1) \models \{\sigma_r(r_1, r_2), p_1, \sigma_v(B, C)\}$. Annotated as a functional property in Figure 4.8(a), p_2 has the value B in Figure 4.8(d), as the first graph has been marked as authoritative beforehand. The value C is discarded. However, σ_v can redefine these criteria, and employ further processing to ensure that property values are equivalent or not.



(a) Semantically Equivalent RDF Molecules

(b) Union

(c) Subproperty

(d) Functional

Figure 4.8: **Merging Semantically Equivalent RDF Molecules**. Applications of a fusion policy σ: (a) semantically equivalent molecules R_1 and R_2 with two ontology axioms; (b) simple union of all triples in R_1 and R_2 without tackling semantics; (c) p_3 is replaced as a subproperty of p_4; (d) p_2 is a functional property and R_1 belongs to the authoritative graph; therefore, literal C is discarded.

- The **equivalent property policy** is triggered when two properties of two molecules are `owl:equivalentProperty`: $\{r_1, p_1, A\}, \{r_2, p_2, B\} + O + \mathit{equivalent}(p_1, p_2) \models \{\sigma_r(r_1,\, r_2),\, \sigma_p(p_1, p_2),\, \sigma_v(A, B)\}$. The authoritative policy selects a property from the authoritative graph, e.g., either p_1 or p_2. By default, the property value is taken from the chosen property. Custom σ_v policies may override these criteria.

- The **equivalent class or entity policy** contributes to the integration process when property values are annotated as `owl:equivalentClass` or `owl:sameAs`, i.e., two classes or individuals represent the same real-world entity, respectively: $\{r_1, p_1, A\}, \{r_2, p_2, B\} + O + \mathit{equiva\text{-}lent}(A, B) \models \{\ \sigma_r(r_1, r_2),\ \sigma_p(p_1, p_2),\ \sigma_v(A, B)\}$. Similarly to the equivalent property case, the value with its corresponding property is chosen from the primary graph. Custom σ_p policies may handle the merging of properties.

The spectrum of possible fusion policies is not limited to the list described above. Fusion policies allow for a flexible management, and for a targeted control of creation of an integrated knowledge graph. These policies vary from naming convention for resources to a fine-grained tuning of desired parameters. Varying a set of applied policies, it is possible to focus on a certain integration aspect

4.3 Properties of our Approach

In this section, we show the two main properties of MINTE: (1) its high adaptability, thanks to the parametrization of its components; and (2) the low complexity, thanks to the efficient algorithms used in at each of the two steps.

4.3.1 High Adaptability

Adaptability is to be understood here as the ability of MINTE to adapt itself efficiently and fast to different semantic interoperability conflicts. Ergo, MINTE is able to fit its integration process according to changes in its environment, e.g., different application domains. Each application domain poses different challenges, for example, in the crime investigation domain the fusion step is only done if the similarity is really high (Threshold 0.9). In contrast, in the job market analysis domain, job adds are fused when they are similar enough (Threshold 0.75). As described in Section 4.2.2, MINTE approach defines five parameters which are described in Table 4.1. Conceptually, it is like having multiple approaches under the same framework, i.e., when the threshold changes the integration approach changes.

Parameter	**Value Type**	**Description**
Ontology	RDFS or OWL	An ontology describing the RDF molecules
Datasets	RDF	The datasets to be integrated by MINTE
Similarity Metric	Function	A similarty function
Threshold	Double	Value to define the threshold when two entities need to be integrated
Fusion Policy	Function	Defines the way two molecules are synthesized

Table 4.1: **MINTE Configuration Parameters**

We empirically evaluated this property by applying MINTE in three domain-specific applications (cf. Chapter 7). During the implementation of these applications, more than 20 different datasets were integrated. MINTE successfully employed the following elements: three fusion policies, i.e., the Union, and the Authoritative policy; a threshold value between 0.0 and 1.0; three similarity metrics, i.e., GADES, Jaccard, SILK rule-based; finally, MINTE has been tested with three different ontologies: OntoFuhsen, SARO, and schema.org.

4.3.2 Low Complexity

Graph Matching Complexity

MINTE receives two sets $\phi(G)$ and $\phi(D)$ of n and k RDF molecules. To estimate the complexity the two most expensive operations have to be analyzed. Table 4.2 gives an overview of the analysis. The complexity of the Dataset Partitioner module depends on the complexity of the chosen similarity measure that has to be applied for nk pairs. The asymptotic approximation thus equals to $O(nk \cdot O(Sim_f))$. The complexity of 1-1 Weighted Perfect Matching component employs the Hungarian algorithm [124] and hence converges to $O(n^3)$. Partitioning and perfect matching are executed sequentially. Therefore, the overall complexity conforms to the sum of complexities, i.e., $O(nk \cdot O(Sim_f)) + O(n^3)$. We thus deduce that the graph matching complexity depends on the complexity of the chosen similarity measure, whereas the lowest achievable order of complexity is limited to $O(n^3)$.

Stage	**Entities Identification**	**Fusion**
Partitioning	$O(nk \cdot O(Sim_f))$	
1-1 Matching	$O(n^3)$	$O(n \cdot O(l^p))$
Overall	$O(nk \cdot O(Sim_f)) + O(n^3)$	$O(n \cdot O(l^p))$

Table 4.2: **Time Complexity**. Results for the steps of Partitioning and Matching, where n, k are the numbers of RDF molecules, $n \geq k$. $l := card(\mathcal{M}_i) + card(\mathcal{M}_j) + card(O)$, i.e., the amount of properties to merge having an ontology O; p is a constant.

Graph Fusion Complexity

Fusion policies resort to axioms, e.g., property hierarchies, functionality, transitivity, disjointness, inverses, symmetry, chains, irreflexivity, that are defined in the OWL 2 RL profile, which is in

	Experiment 1: People		**Experiment 2: People**		**Experiment 3: Drugs**	
	DBpedia D1	DBpedia D2	DBpedia	Wikidata	DBpedia	DrugBank
Molecules	500	500	20,000	20,000	1,568	1,568
Triples	17,951	17,894	1,421,604	855,037	398,043	517,023

Table 4.3: **Benchmark Description**. RDF datasets used in the evaluation

turn based on the DLP logic [125]. Reasoning in OWL 2 RL is proven to be polynomial [126]. Therefore, given n identified pairs, a number of $l := card(\mathcal{M}_i) + card(\mathcal{M}_j) + card(O)$ of properties in the compared molecules and ontology O, then the fusion complexity conforms to $O(n \cdot O(l^p))$, where p is a constant, i.e., polynomial complexity.

4.4 Experimental Studies

The MINTE approach exploits the semantics encoded in RDF molecules at each step of its pipeline. To answer research question 1 (cf. Section 1.3) stated in this thesis, we evaluate the effectiveness of MINTE in solving the integration problem between RDF graphs. We conducted three experiments evaluating different types of heterogeneity on the schema, properties, and value levels, using RDF graphs from DBpedia, Wikidata, and Drugbank. We address the following questions:

- **Q1:** Is MINTE capable of integrating diverse RDF graphs effectively?
- **Q2:** How does a similarity function affect the effectiveness of the MINTE integration technique?

4.4.1 Metrics and Settings

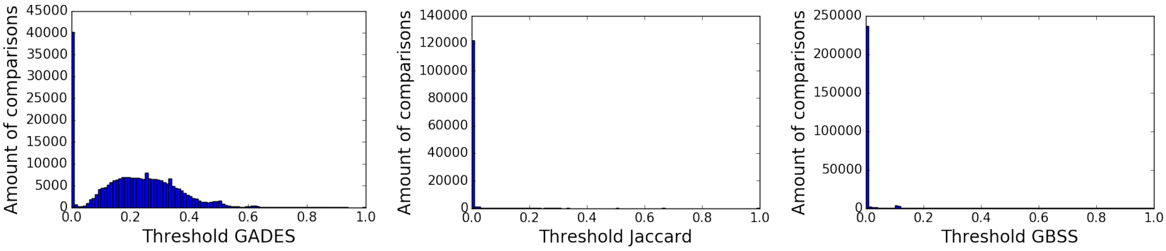
MINTE is implemented in Python 2.7.10. The experiment was executed on a Ubuntu 14.04 (64 bits) machine with CPU: Intel Xeon E5-2650 2.3 GHz (4 physical cores) and 32 GB RAM. We evaluated three similarity functions in the MINTE pipeline: $\mathcal{GADES}$ [7], Semantic Jaccard (SemJaccard) [127], and GBSS [128]. $\mathcal{GADES}$ relies on semantic description encoded in ontologies to determine relatedness. $\mathcal{GADES}$ examines both hierarchy similarity, i.e., graph neighbourhoods, and string similarity. SemJaccard is an extension of Jaccard similarity metric adjusted for supporting reasoning and materialization. That is, comparing entities from different vocabularies, SemJaccard requires the materialization of implicit knowledge and mappings instead of direct triple sets comparison as plain Jaccard does. Deduced facts increase the possible intersection of triples and raise the similarity score. Finally, GBSS[7] is a similarity function that is tailored only for DBpedia vocabularies.

Although each experiment has different datasets and gold standards, we use the same metrics for all the experiments. We measure *Precision*, *Recall* and *F-measure*. *Precision* is the fraction of RDF molecules that has been identified and integrated by MINTE (M) that intersects with the Gold Standard (GS), i.e., $Precision = \frac{|M \cap GS|}{|M|}$. *Recall* corresponds to the fraction of the identified similar molecules in the Gold Standard, i.e., $Recall = \frac{|M \cap GS|}{|GS|}$. *F-measure* is the

[7] https://github.com/chrispau1/SemRelDocSearch

	GADES			SemJaccard			GBSS		
γ	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
NT	0.810	0.814	0.810	0.770	0.784	0.777	0.466	0.466	0.466
P95	0.836	0.808	0.822	0.784	0.784	0.784	0.906	0.462	0.612
P97	0.840	0.808	0.824	0.909	0.782	0.841	0.924	0.46	0.614
P99	0.857	0.758	0.804	0.910	0.770	0.834	0.936	0.382	0.543

Table 4.4: **Experiment 2: MINTE Effectiveness on DBpedia.** Values of γ correspond to percentiles: 95, 97, and 99, and No-Threshold (NT). MINTE exhibits the best performance for semantic similarity functions, e.g., *GADES* and SemJaccard



(a) $\mathcal{GADES}$ (b) SemJaccard (c) GBSS

Figure 4.9: **Histogram of the Similarity Scores** between $\mathcal{GADES}$, SemJaccard, and GBSS for DBpedia Molecules with different threshold values

harmonic mean of *Precision* and *Recall*. Precision and Recall equally contribute to the final score; therefore we compute the *F1* metric.

4.4.2 Integrating RDF Molecules from DBpedia

The goal of this experiment is to evaluate the MINTE approach on RDF graphs that share the same vocabulary, while the RDF molecules have different properties.

Benchmark: We extracted 500 molecules[8] of type Person from the live version of DBpedia (Released on July 2016). Based on the original RDF molecules we created two sets of molecules by randomly deleting or editing triples in the two sets. Table 4.3 (Experiment 1) provides basic statistics on the benchmark.

Baseline: The gold standard includes the original DBpedia person entities and corresponds to the idealized RDF graph F. The fusion policy is set to the default one, i.e., the *Union policy*.

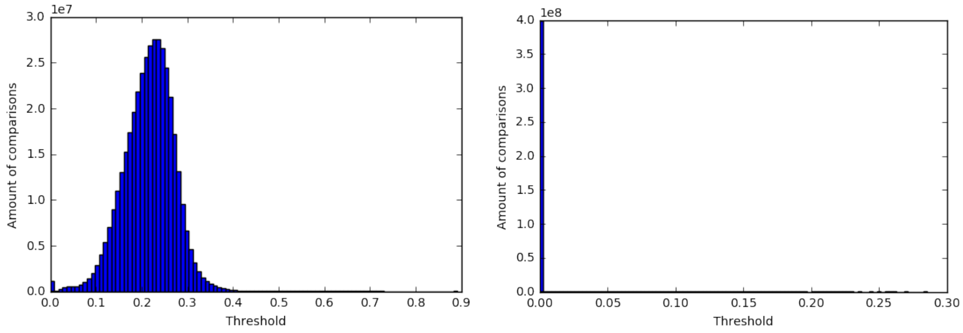
We evaluate MINTE with $\mathcal{GADES}$, SemJaccard, and GBSS on datasets D_1 and D_2 presented in Table 4.3. RDF molecules are described using the DBpedia vocabulary, and each molecule has only one corresponding semantically equivalent molecule. Table 4.4 shows the results obtained from the integration of DBpedia molecules.

MINTE exhibits high values of precision and recall for SemJaccard and $\mathcal{GADES}$, and the best F-Measure value is achieved when MINTE utilizes SemJaccard and only similarity values above the 97th percentile are considered; histograms are reported in Figure 4.9. Because no schema heterogeneity exists in D_1 and D_2, and the DBpedia ontology encodes a large number of class and property hierarchies, MINTE is able to accurately integrate RDF molecules.

[8] Datasets are available at: https://github.com/RDF-Molecules/Test-DataSets

	$\mathcal{GADES}$			SemJaccard		
γ	Precision	Recall	F-Measure	Precision	Recall	F-Measure
NT	0.76	0.76	0.76	0.253	0.253	0.253
P95	0.836	0.588	0.69	0.253	0.253	0.253
P97	0.861	0.537	0.661	0.253	0.253	0.253
P99	0.913	0.39	0.547	0.282	0.252	0.266

Table 4.5: **MINTE Effectiveness on DBpedia and Wikidata Molecules** Values of γ correspond to the percentiles: 95, 97, and 99, and No-Threshold (NT). MINTE exhibits a better performance in $\mathcal{GADES}$, while SemJaccard is affected by the heterogeneity of DBpedia and Wikidata vocabularies



(a) $\mathcal{GADES}$

(b) SemJaccard

Figure 4.10: **Histogram of the similarity scores** of $\mathcal{GADES}$ and SemJaccard for DBpedia and Wikidata datasets with different threshold values

4.4.3 Integrating DBpedia and Wikidata RDF Molecules

The goal of this experiment is to evaluate MINTE approach on RDF graphs that contain semantically equivalent entities but are annotated with different vocabularies, namely DBpedia and Wikidata.

Benchmark: Table 4.3 (Experiment 2) describes the datasets containing 20,000 molecules of type Person extracted from the live version of DBpedia (July 2016) and Wikidata.

Baseline: The gold standard includes the `owl:sameAs` links between entities from DBpedia and Wikidata. The fusion policy is set to the default *Union* policy.

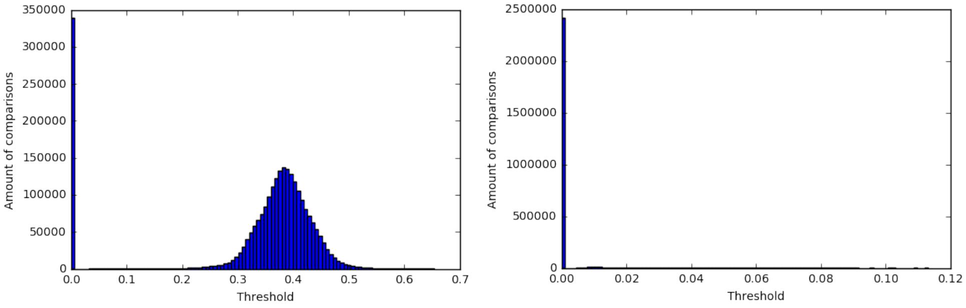
We evaluate how MINTE performs when integrating datasets described with different vocabularies. Table 4.5 contains the results of MINTE using the SemJaccard and $\mathcal{GADES}$ similarity measures. We observe that MINTE exhibits the best behavior when $\mathcal{GADES}$ is utilized, i.e., the maximal F-Measure is 0.76 in comparison to 0.266 obtained by SemJaccard. $\mathcal{GADES}$ considers semantics and is able to leverage equivalence and subsumption relations between entities in RDF graphs. Thus, even when the molecules are described with different vocabularies, $\mathcal{GADES}$ is able to detect relatedness between RDF molecules. However, SemJaccard does not utilize this semantics and therefore, it produces worse results even in high percentiles (cf. Figure 4.10).

4.4.4 Integrating RDF Molecules from DBpedia and Drugbank

The goal of this experiment is to evaluate MINTE against RDF graphs annotated with different vocabularies. In the third experiment we compare Drug entities.

	$\mathcal{GADES}$			SemJaccard		
γ	Precision	Recall	F-Measure	Precision	Recall	F-Measure
NT	0.749	0.749	0.749	0.854	0.854	0.854
P95	0.882	0.502	0.64	0.854	0.854	0.854
P97	0.88	0.409	0.558	0.854	0.854	0.854
P99	0.859	0.191	0.313	0.99	0.851	0.915

Table 4.6: **MINTE Effectiveness on DBpedia and Drugbank** Values of γ correspond to the percentiles: 95, 97, and 99, and No-Threshold (NT). MINTE exhibits a better performance with SemJaccard because the heterogeneity between DBpedia and Drugbank vocabularies is addressed by hand-crafted mappings



(a) $\mathcal{GADES}$

(b) SemJaccard

Figure 4.11: **Histogram of the similarity scores** between $\mathcal{GADES}$ and SemJaccard similarity functions for the Drugs dataset with different threshold values

Benchmark: contains 1568 molecules of type Drug extracted from the live version of DBpedia (July 2016) and Drugbank. Table 4.3 (Experiment 3) shows details of the involved datasets.

Baseline: The gold standard includes the links between DBpedia and Drugbank entities and corresponds to the idealized RDF graph F. The fusion policy is set to the default *Union* policy in each experiment conducted.

Table 4.6 shows the results of MINTE with SemJaccard and $\mathcal{GADES}$. Contrarily to the previous experiment, heterogeneity of vocabularies is addressed by mappings between DrugBank properties and DBpedia properties. Some mappings are already described in the datasets by `owl:sameAs` and `owl:equivalentProperty` axioms, while other mappings have been hand-crafted for this experiment. The mappings produce more materialized triples that in turn increase the performance of SemJaccard. Varying the threshold value, MINTE manages to achieve 0.915 F-Measure for the 99th percentile (cf. Table 4.6).

4.4.5 Discussion of Observed Results

Based on the three experiments considering Precision, Recall, and F-Measure, we can positively answer **Q1**, i.e., MINTE is capable to integrate semantically equivalent RDF molecules to create an integrated RDF graph. We can also observe that the accuracy of MINTE is indeed affected by the behavior of the studied similarity measures, as shown in the Tables 4.4, 4.5, and 4.6. Therefore, these observed results allow us to answer **Q2**: a semantic similarity function tends to produce more precise and reliable results than non-semantic ones.

4.5 Summary

After many approaches and techniques to integrate heterogeneous data using semantic technologies, the integration of semantic equivalent entities from heterogeneous Web sources in a single pipeline remained unfulfilled. In this chapter, we presented MINTE, the first “RDF Molecule-based Integration Technique” for integrating semantically equivalent RDF molecules from Web sources into a single RDF graph. MINTE follows a two-fold approach where first semantically equivalent RDF molecules are identified, and then, semantically equivalent RDF molecules are merged. MINTE may utilize different similarity measures to decide whenever two RDF molecules are equivalent. Furthermore, MINTE resorts to different fusion policies to merge semantically equivalent RDF molecules. We show that the MINTE computation complexity is in the order of polynomial time, therefore, MINTE can be effectively applied for integrating semantically equivalent RDF molecules from different Web sources. The behavior of MINTE was empirically studied on three real-world RDF graphs and on three similarity measures. Observed results suggest that MINTE is able to effectively identify and merge semantically equivalent entities, and is empowered by the semantics encoded in ontologies and can exploit similarity measures. MINTE defines a set of input parameters making the integration process flexible and applicable to different application domains.

CHAPTER 5

A Semantic Similarity Framework for Knowledge Integration

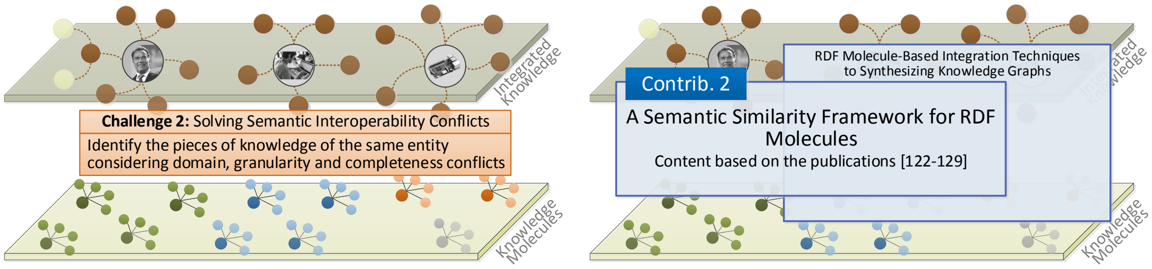
In this chapter, we focus on the problem of determining relatedness among RDF molecules at integration time. The content of this chapter is based on the publications [122, 129]. A semantic similarity metric is a key building block of MINTE, the RDF molecule-based integration technique we defined in Chapter 4. Thus, in this chapter, we present a semantic similarity framework that includes two semantic similarity metrics adapted to work with RDF molecules, i.e., GADES and MateTee. The semantic similarity metrics included in the framework help to improve the performance of the MINTE's integration process. The results of this chapter provide an answer to the following research question:

RQ2: How can semantic similarity metrics facilitate the process of integrating data collected from heterogeneous web sources?

To solve the problem of determining relatedness between entities several similarity metrics have been proposed. Traditional similarity metrics translate the entity's properties into a mathematical representation, where the comparison is easily measurable. Nevertheless, we focus our study and analysis on similarity metrics that exploit the semantics encoded in RDF molecules (what we call a semantic similarity metric). Figure 5.1(a) shows the main problems to compare entities extracted from data spread over heterogeneous web sources.

First, we motivate the problem of determining relatedness among entities in knowledge graphs using a practical example in Section 5.1. Then we show an empirical evaluation of the impact on the integration process accuracy cause by semantic similarity metrics. We adapt two similarity metrics to work with RDF molecules, Jaccard (no semantic, Section 5.2.1) and GADES [7] (semantic, Section 5.2.2). Finally, we compare the performance of the two approaches via an empirical study and we show the results in Section 5.2.3.

Besides the analysis of the impact of state-of-the-art semantic similarity metrics, in Section 5.3 we propose an new similarity metric for RDF molecules based on embeddings. Section 5.3.1 describes the details of embedding concept, this is required to understand the proposed solution. Section 5.3.2 defines the problem, the proposed solution, and the MateTee architecture. Section 5.3.3 reports on the empirical evaluation. Finally, Section 5.4 presents the closing remarks of this chapter. In summary, the contributions described in this chapter are the following:


Integrated Knowledge
Challenge 2: Solving Semantic Interoperability Conflicts
Identify the pieces of knowledge of the same entity considering domain, granularity and completeness conflicts
Knowledge Molecules
RDF Molecule-Based Integration Techniques to Synthesizing Knowledge Graphs
Contrib. 2
A Semantic Similarity Framework for RDF Molecules
Content based on the publications [122-129]
Integrated Knowledge
Knowledge Molecules

(a) Problems tackled in this chapter

(b) Contributions described in this chapter

Figure 5.1: **Challenges and Contributions:** This chapter focuses on the problem of identifying semantically equivalence entities from different web sources, and proposes a semantic similarity framework for RDF molecules to solve this problem.

- An empirical evaluation of the impact on the integration task using a semantic similarity metric, i.e., GADES [7].
- An *end-to-end* approach named MateTee that is able to compute similarity values among entities in a knowledge graph. MateTee is based on TransE, which utilizes the gradient descent optimization method to learn a *features representation* of the entities automatically.
- An extensive empirical evaluation on existing benchmarks and state-of-the-art showing MateTee behavior. Results indicate the benefits of using embeddings for determining relatedness among entities in a knowledge graph. MateTee and experimental studies are publicly available[1].

5.1 The Need for a Semantic Similarity Framework

The semantic representation of the data in RDF helps in the endeavor of automatically solving data-driven oriented tasks, providing as result, more useful and meaningful services from such big and heterogeneous data [26]. Particularly, the tasks affected by a good similarity metric between data entities are: semantic data integration of heterogeneous data, or entity linking and clustering. The future of the Web of Data and the Web of Things brings even more heterogeneity and larger datasets. Streaming data coming at high rates need to be processed on-demand, all of which only increases the need of automation in the process of creation and processing of semantics. In the case of knowledge graphs, we are referring to classification of entities in a set of classes, and prediction (or discovery) of new relations between entities, i.e., RDF triples. Consider a knowledge graph in Figure 5.2. Nodes of the same color indicate they share the same properties, while nodes of different colors differ in at least one property. Determining relatedness among same-colored nodes, e.g., Camilo with Diego, requires to compare, in a 1-1 fashion, values of each property of those entities and aggregate the results. This computation can be done as Camilo and Diego have the same set of properties, i.e., *Child_of* and *Birth_Place*. Contrary, if entities have different properties, i.e., they are on different colors, the problem is to measure their relatedness considering the complete set of properties of both nodes while is not possible to use the 1-1 approach, e.g., Germany and Camilo. Moreover, whenever entities are compared

[1] https://github.com/RDF-Molecules/MateTee

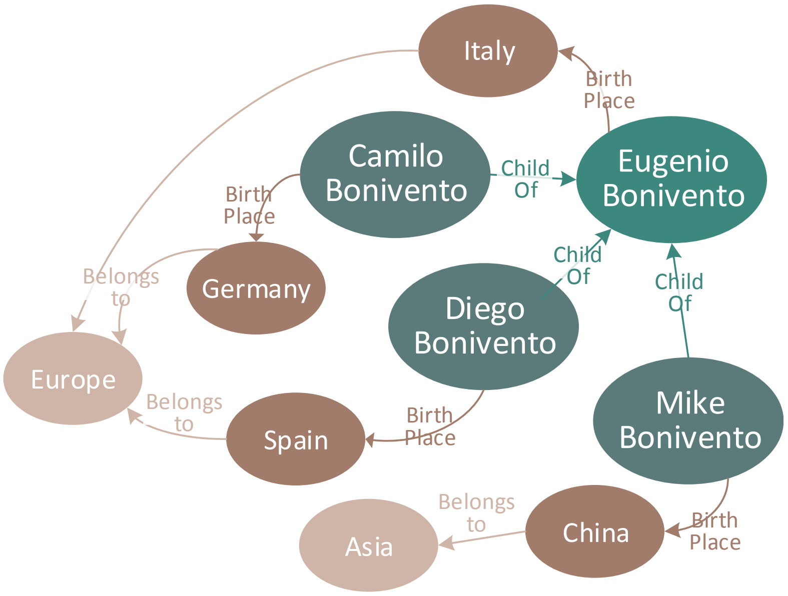


Figure 5.2: **Motivating Example.** A portion of a knowledge graph describing relationships among persons and the places where they have been born. There exist different types of relations and multiple connectivity patterns among the entities.

in terms of their *neighborhoods* and *reachable* nodes, Camilo should be more similar to Diego than to Mike, as Diego and Camilo are from Europe, while Mike is from China.

These difficulties come inherently with the multi-relational datasets. In relational data tables, all elements have the same properties, i.e., columns, and therefore, the similarity computation is performed aggregating a 1-1 similarity value between each pair of the properties. With multi-relational data, nodes need to be made *comparable*, which means that they all must have the same set of properties or features. This can be done manually, *handcrafting* the features, and creating a list of them for each node, based on previous knowledge of the specific field or domain of the data. These sets of features will be regarded as a new representation of the nodes in the knowledge graph. Then, these sets of features can be compared, again, in a 1-1 fashion. The problem is that manual creation of the features requires deep domain knowledge, not to mention it is error-prone and time consuming. Thus, to solve these problems a similarity measurement approach that automatically creates a canonical entity representations is required.

Data management and Artificial Intelligence approaches play an important role on the task of knowledge graph data analysis. Machine Learning (ML), mostly in its supervised flavor, aims to give machines the capability to learn by examples, essentially, labeled data. ML field has achieved promising results with sophisticated techniques, such as Kernel methods or Deep Learning models. Furthermore, the Semantic Web, and in general, all the available knowledge graphs such as DBpedia or Yago, have been built with a tremendous effort of the scientific community having the main objective of making the data understandable not only by humans but also by computers. Structured data facilitates the tasks of data integration, relations or associations discovery, as presented by Bordes et al. 2013 with TransE [102]. On one hand, we have an immense amount of available knowledge facts, encoded as structured data in knowledge graphs, and on the other, we have the Machine Learning boom and techniques able to have access to Big Data sets, for two main tasks: classification and link prediction.

5.2 Semantic Similarity Metrics for RDF Molecules

In this section, we present two metrics for RDF molecules, the classic (no semantic) Jaccard metric is used as the baseline of comparison, and GADES a state-of-art semantic similarity metric. We explain the re-definition of Jaccard and GADES to work on RDF molecules of data. We present an empirical performance analysis of both approaches on the task of identifying the similarity among RDF molecules from heterogeneous web sources for integration.

5.2.1 Jaccard Similarity for RDF Molecules

We use *Jaccard distance* to compute a similarity score of two molecules, and it is defined as follows: Let A be an RDF molecule with a set T_1 of n properties and values (i.e. $|T_1| = n$), and let B be an RDF molecule with a set T_2 of k properties and values (i.e., $|T_2| = k$). The Jaccard similarity is then computed as:

$$Jaccard(A, B) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$$

The intersection set contains only those pairs of $\langle property, val \rangle$ that are present in both T_1 and T_2. The union set contains all unique $\langle property, val \rangle$ pairs.

5.2.2 GADES for RDF Molecules

GADES[2] [7] is a semantic similarity metric used to compare entities in a knowledge graph. GADES considers three different aspects: the class hierarchy, the neighbors of the entities, and the specificity of the entities. Thus, GADES is defined as a combination of three similarity values Sim_{hier}, $\text{Sim}_{\text{neigh}}$ and Sim_{spec}. These similarity values can be combined with different T-Norms as the product or the average depending on the domain. In the case of the RDF molecules we define GADES as:

$$GADES(A, B) = \frac{\text{Sim}_{\text{hier}}(A, B) + \text{Sim}_{\text{neigh}}(A, B)}{2}$$

Hierarchical similarity. Given a knowledge graph G, the hierarchy is inferred by the set of hierarchical edges. Hierarchical edges are a subset of knowledge graph edges whose property names refer to a hierarchical relation, e.g., *rdf:type* or *rdfs:subClassOf*. In the case of DBpedia and according to Lam et al. [130], the Wikipedia Category Hierarchy is used to determine the hierarchical similarity between two entities. Thus, the hierarchy is induced by relations *skos:broader* and *dc:subject*. Given this hierarchy, d_{tax} [131] is used by GADES to measure the hierarchical similarity between two entities.

Neighborhood similarity. The neighborhood of an entity e in a RDF molecule M is defined as the set of property-object pairs included in the triples of the molecule $N(e) = \{(p, o) | (s, p, o) \in M\}$. Thus, there are two types of neighbors: URIs representing entities and literals representing attributes. This definition of neighborhood allows for considering together the neighbor entity and the relation type of the edge. GADES uses the knowledge encoded in the relation and class

[2] The adaptation of GADES is a joint work with Ignacio Traverso Ribón, a Ph.D. student at the Karlsruher Institut für Technologie (KIT). My contributions include the preparation and implementation of a REST service for RDF molecules.

hierarchies of the knowledge graph to compare two pairs $n_1 = (p_1, o_1)$ and $n_2 = (p_2, o_2)$. The similarity between two pairs n_1 and n_2 is computed as follows:

- If o_1 and o_2 are URIs, GADES uses a hierarchical similarity measure between URIs:

$$\mathrm{Sim}_{\mathrm{pair}}(n_1, n_2) = \frac{\mathrm{Sim}_{\mathrm{hier}}(o_1, o_2) + \mathrm{Sim}_{\mathrm{hier}}(p_1, p_2)}{2}$$

- If o_1 and o_2 are literals, GADES uses the Jaro-Winkler similarity measure between literals:

$$\mathrm{Sim}_{\mathrm{pair}}(n_1, n_2) = \frac{\mathrm{Sim}_{\text{Jaro-Winkler}}(o_1, o_2) + \mathrm{Sim}_{\mathrm{hier}}(p_1, p_2)}{2}$$

In order to maximize the similarity between two neighborhoods, GADES combines pair comparisons as: $\mathrm{Sim}_{\mathrm{neigh}}(e_1, e_2) = \frac{\sum_{i=0}^{|N(e_1)|} \max_{n_x \in N(e_2)} \mathrm{Sim}_{\mathrm{pair}}(n_i, n_x) + \sum_{j=0}^{|N(e_2)|} \max_{n_y \in N(e_1)} \mathrm{Sim}_{\mathrm{pair}}(n_j, n_y)}{|N(e_1)| + |N(e_2)|}$

5.2.3 Empirical Studies

With the following configuration, we empirically study the impact of a semantic similarity metric, i.e., GADES on the task of integrating data from web sources (cf. Section 1.3, RQ2). By assessing the following research questions, we evaluate the impact of GADES on the integration problem of RDF molecules:

- **Q1:** Can the semantic similarity metric implemented in *GADES* integrate data in a knowledge graph more accurately than Jaccard a no-semantic similarity metric?
- **Q2:** Is the accuracy of the molecule-based integration technique implemented in *MINTE* impacted by the similarity metric used during integration process?

Gold Standard (GS): The ground truth dataset was extracted from the live version of DBpedia (July 2016). We created two subsets of the ground truth to evaluate the scalability of the similarity metrics. The first GS contains 500 molecules of type Person[3], i.e., 500 subjects with all available properties and their values. The overall number of triples is 20,936. The second GS contains 20,000 molecules of the type Person, which results in 829,184 triples. The Gold Standards are used to compute precision and recall during the evaluation.

Test Datasets (TS): The molecules from the Gold Standard with their properties and values were randomly split among two test datasets. Each triple is randomly assigned to one or several test datasets. The selection process takes two steps: 1) a number of test datasets to copy a triple to is chosen randomly under a uniform distribution; 2) the chosen number is used as a sample size to randomly select particular test datasets to write a triple. URIs are generated specifically for each test dataset. Eventually, each test dump contains a subset of the properties in the gold standard. Each subset of properties of each molecule is composed randomly using a uniform distribution. A small tweak was made as to the first Gold Standard in order to make both test datasets contain 500 molecules. Nevertheless, properties were still assigned randomly to each test dataset. Tables 5.1 and 5.2 provide additional statistics on the data sources.

[3] http://dbpedia.org/ontology/Person

	DataSet1	DataSet2	Gold
Size (MB)	2.3	2.3	3.2
RDF Molecules	500	500	500
Triples	14,692	14,705	20,936

Table 5.1: **Description of Datatsets**. 500 molecules

	DataSet1	DataSet2	Gold
Size (MB)	86.1	85.9	124
RDF Molecules	13,242	13,391	20,000
Triples	553,059	552,425	829,184

Table 5.2: **Description of Datatsets**. 20,000 molecules

Metrics: We measure the behavior of *GADES* in terms of the following metrics:
a) **Precision** is the fraction of RDF molecules identified and integrated by *GADES* (M) that intersects with the Gold Standard (GS).

$$Precision = \frac{|M \cap GS|}{|M|}$$

b) **Recall** is the cardinality of the intersection of molecules (M) integrated and Gold Standard (GS), divided by that of the Gold Standard (GS).

$$Recall = \frac{|M \cap GS|}{|GS|}$$

c) **F-measure** is the harmonic mean of *Precision* and *Recall.*

Implementation: Experiments were run on a Windows 8 machine with an Intel i7-4710HQ 2.5 GHz CPU and 16 GB 1333 MHz DDR3 RAM. We implemented *GADES* and the Jaccard similarity metric in Scala and Java. Further, the transformation of the RDF molecules was implemented using Jena in Java 1.8. The *GADES* framework, and the test sets evaluated in this experiment are publicly available.[4]

Discussion: With this experiment, we answer our research questions **Q1** and **Q2**. *GADES* is run on the two test sets of different sizes to calculate the similarity among molecules with a triple-based approach implemented by Jaccard and a molecule-based one implemented by GADES. We compute Precision, Recall, and F-measure according to the Gold Standard. Table 5.3 reports on the values of these metrics for 500 molecules, Table 5.4 contains the values for 20,000 molecules. Jaccard demonstrates lower performance on both datasets as it relies just on the particular properties of the RDF molecule. Jaccard does not utilize semantics encoded in the knowledge graph and cannot be used as a 'black box' to compute the similarity between arbitrary sets of molecules without prior knowledge of the data model of those RDF molecules.

On the other hand, GADES might be used as such a 'black box' as it does not require any metadata or knowledge of the schema. Nevertheless, the performance depends on the threshold parameter. As a simple sets-based approach, the performance (precision, recall, and F-Measure)

[4] https://github.com/LiDaKrA/RDF-Molecules-Experiment

Precision										
	T0.0	T0.1	T0.2	T0.3	T.0.4	T0.5	T0.6	T0.7	T0.8	T0.9
Jaccard	0.77	0.84	0.55	0.43	0.45	0.45	0.62	0.4	0.4	0.4
GADES	0.81	0.86	0.86	0.86	0.86	0.86	0.86	**0.87**	0.83	**0.87**
Recall										
	T0.0	T0.1	T0.2	T0.3	T.0.4	T0.5	T0.6	T0.7	T0.8	T0.9
Jaccard	0.77	0.5	0.1	0.05	0.03	0.03	0.01	0.004	0.004	0.004
GADES	**0.81**	**0.81**	**0.81**	**0.81**	**0.81**	**0.81**	0.77	0.59	0.26	0.07
F-Measure										
	T0.0	T0.1	T0.2	T0.3	T.0.4	T0.5	T0.6	T0.7	T0.8	T0.9
Jaccard	0.77	0.63	0.17	0.1	0.06	0.06	0.02	0.008	0.008	0.008
GADES	0.81	**0.84**	**0.84**	**0.84**	**0.84**	**0.84**	0.81	0.70	0.40	0.13

Table 5.3: **Effectiveness of *GADES* on 500 RDF molecules**. Jaccard triple-based integration vs GADES semantic integration approach using different thresholds (T). Highest values of Recall and F-measure are highlighted in bold.

Precision										
	T0.0	T0.1	T0.2	T0.3	T.0.4	T0.5	T0.6	T0.7	T0.8	T0.9
Jaccard	0.72	0.77	0.44	0.34	0.37	0.36	0.27	0.21	0.21	0.21
GADES	0.76	**0.80**	**0.80**	0.79	0.79	0.79	0.79	0.76	0.70	0.65
Recall										
	T0.0	T0.1	T0.2	T0.3	T.0.4	T0.5	T0.6	T0.7	T0.8	T0.9
Jaccard	0.72	0.42	0.09	0.05	0.02	0.02	0.01	0.01	0.01	0.01
GADES	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	**0.76**	0.68	0.46	0.22	0.06
F-Measure										
	T0.0	T0.1	T0.2	T0.3	T.0.4	T0.5	T0.6	T0.7	T0.8	T0.9
Jaccard	0.72	0.54	0.15	0.08	0.04	0.04	0.02	0.02	0.02	0.02
GADES	0.76	**0.78**	**0.78**	0.77	0.77	0.77	0.73	0.57	0.33	0.11

Table 5.4: **Effectiveness of *GADES* on 20,000 RDF molecules**. Jaccard triple-based integration vs GADES semantic integration approach using different thresholds (T). Highest values of Recall and F-measure are highlighted in bold.

of the Jaccard similarity quickly decreases with higher thresholds. On low thresholds only one or two common triples between molecules are sufficient to mark the molecules as similar even though other properties and values are different. Higher thresholds increase the necessary amount of common triples to classify molecules as similar. However, GADES leverages higher semantic abstraction layers involving hierarchies and neighborhoods. GADES is capable of maintaining stable performance and quality on thresholds up to 0.7 despite the size of the datasets. The drop at higher thresholds is explained by insufficient amounts of common triples which serve as a basis for materialization of class hierarchy, property hierarchy, and neighborhood. **Q1** is therefore confirmed, as one can vary the quality of interlinking in a wide range, whereas in the triple-based approach the quality is always constant. The accuracy of the molecule-based integration approach (**Q2**) is indeed affected by a similarity measure and its parameters as shown in the Table 5.3 and Table 5.4.

5.3 A Semantic Similarity Metric Based on Translation Embeddings

In this section, we present MateTee, a similarity approach that relies on embedding the original knowledge graph into a vector space in order to make all entities comparable. Similarity values among embeddings are measured based on any distance metric defined for vector spaces, e.g., Euclidean distance. Large knowledge graphs e.g., DBpedia or Wikidata, are created with the goal of providing structure to unstructured or semi-structured data. Having these special datasets constantly evolving, the challenge is to utilize them in a meaningful, accurate, and efficient way. Further, exploiting semantics encoded in knowledge graphs e.g., class and property hierarchies, provides the basis for addressing this challenge and producing a more accurate analysis of knowledge graph data. Thus, we focus on the problem of determining relatedness among entities in knowledge graph, which corresponds to a fundamental building block for any semantic data integration task. We devise MateTee, a semantic similarity metric that combines the *gradient descent* optimization method with semantics encoded in ontologies, to precisely compute values of similarity between entities in knolwedge graphs. We empirically study the accuracy of MateTee with respect to state-of-the-art methods. The observed results show that MateTee is competitive in terms of accuracy with respect to existing methods, with the advantage that background domain knowledge is not required.

5.3.1 Background: Translation Embeddings

MateTee determines relatedness between entities in Knowledge Graphs based on Encoding Generation methods such as TransE [102]. MateTee combines the gradient descent optimization method (used in TransE) with the explicit knowledge encoded in the ontologies of a knowledge graph. MateTee is based on TransE [102], acronym of *Translation Embeddings*, presented by Bordes et al. 2013. TransE tackles the problem of embedding a knowledge graph into a low dimensional vector space (called embedding space) for subsequent prediction or classification objectives, e.g., predict missing edges. The core of TransE is to learn the embeddings of entities in a way that similar entities in the knowledge graph should be also close in the embedding space. Additionally, dissimilar entities in the knowledge graph should be also far in the embeddings space. Learning the embeddings is done by analysing the *connectivity patterns* between entities in a knowledge graph, and then encoding these patterns into their vector representation, i.e., their embeddings. The optimization technique *Stochastic Gradient Descent* is executed to compute this encoding. Modeling RDF triples in the embedding space with relations as **translations** is the core contribution of TransE. The basic idea behind translation-based model is the following:

$$Subject + Translation \approx Object$$

TransE aims at minimizing the error when summing up the distance d between the embeddings of the $Subject{+}Translation$ pair and the embedding of the $Object$. Stochastic Gradient Descent (SGD) meta-heuristic allows for learning entity embeddings by minimizing the error defined as the sum of the distances d of all the triples in the knowledge graph. A global minimum cannot be ensured because SGD depends on a *randomly* selected start position of the *descent*. The random initialization procedure followed by TransE is presented in detail at [132]. Figure 5.3 illustrates the intuition of this approach.

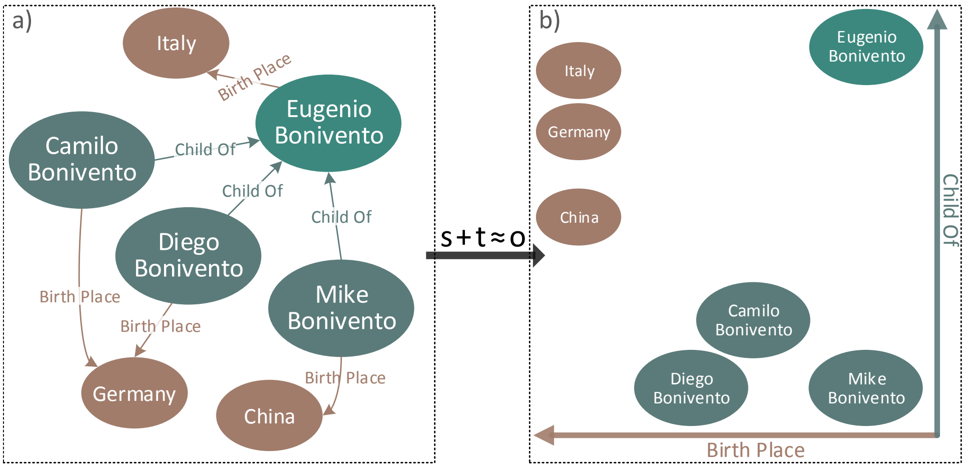


Figure 5.3: **TransE approach intuition.** (a) An RDF Knowledge Graph where similar entities are in the same color; (b) Clusters of entities in the embedding space. Entities of the same color are close to each other in the identified cluster.

5.3.2 The MateTee Approach

MateTee focuses on measuring the similarity between any pair of entities belonging to an input RDF Knowledge Graph. Measuring the similarity between entities is an important phase for any data integration problem, and for most machine learning tasks, e.g., clustering of nodes, or link prediction in knowledge graphs. The main problem for computing the similarity of RDF knowledge graphs is that not all the nodes have the same properties, therefore, a 1-1 comparison at property level cannot be performed. State-of-the-art methods like GADES [7] perform a semantic analysis of the entities based on multiple aspects, i.e., 1-hop neighborhood, class hierarchy of the subjects/objects, class hierarchy of the properties, and mixtures of them. This analysis relies on domain knowledge and user expertise about the provenance of the data, e.g., GADES requires a good design of the hierarchy of classes and properties.

To overcome this problem, MateTee embeds an RDF knowledge graph into a vector space, once all the entities are represented as vectors with same dimensionality, it uses any common distance metric to calculate their similarity values. MateTee relies on finding a vector representation of graph entities to produce the similarity value. For this, MateTee utilizes TransE [102], a method based on Stochastic Gradient Descent that encodes the connectivity patterns of the entities into a low-dimensional embedding space. TransE ensures that similar nodes in the RDF graph are close in the embedding space, while dissimilar nodes in the graph are distant in the embedding space. By using TransE approach, MateTee aims to calculate similarity values as close as possible to the *ground truth*: values accepted by the scientific community because they were calculated manually with deep domain expertise, e.g, Sequence Similarity in the Gene Ontology domain. Formally, MateTee can be defined as:

Definition 3 (MateTee Embedding) *Given a knowledge graph $G = (V, E)$ composed by a set T of RDF triples, where $V = \{s \mid (s,p,o) \in T\} \cup \{o \mid (s,p,o) \in T\}$ and $E = \{p \mid (s,p,o) \in T\}$, MateTee aims to find a set M of embeddings of each member of V, such that:*

$$\arg \min_{\mathbf{m_1},\mathbf{m_2} \in \mathbf{M}} Error(M) = \arg \min_{\mathbf{m_1},\mathbf{m_2} \in \mathbf{M}} \sum_{\mathbf{m_1},\mathbf{m_2} \in \mathbf{M}} |S_1(\mathbf{m_1},\mathbf{m_2}) - S_2(\mathbf{m_1},\mathbf{m_2})|$$

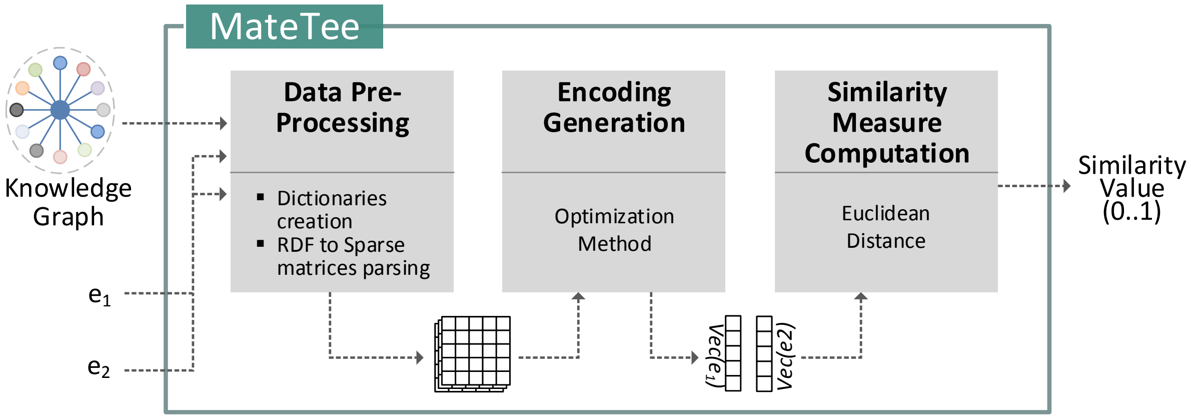


Figure 5.4: **The MateTee Architecture.** MateTee receives as input an RDF Knowledge Graph, and entities e_1 and e_2 from the knowledge graph. MateTee outputs a similarity value between e_1 and e_2 according to the connectivity patterns found in knowledge graph. A pre-processing step allows for the transformation of a knowledge graph into a matrix-based representation. Then, n-dimensional embeddings are generated. Finally, values of similarity are computed.

where S_1 is a similarity metric computed using any distance measure defined for vector spaces, e.g., Euclidean distance, and S_2 is a similarity value given by the Gold Standards. The Gold Standards are the values considered as ground truth.

The MateTee Architecture

Figure 5.4 depicts the *end-to-end* MateTee architecture. MateTee receives as input an RDF Knowledge Graph, and entities e_1 and e_2 belonging to the knowledge graph. The objective of the complete process is to calculate the similarity value between e_1 and e_2.

Data Preprocessing

The first step is to **Pre-Process** the original data in order to transform it into the format required by the optimization method. As the optimization methods are numeric based, we need a numerical representation of the data. In other words, the string-based triples coming as input must be translated into a numeric format, usually sparse matrices. The implementation of TransE employs three sparse matrices: one representing the Objects, another for the Subjects, and a third one for the Translations. The matrices have as many columns as RDF triples are in the original knowledge graph, and as many rows as entities, i.e., number of Subjects + number of Translations + number of Objects. Note that if a Subject appears also as Object in another RDF triple, it is considered as one. Moreover, in order to map the original entities to their respective *encodings*, i.e., embeddings, dictionaries need to be created. Dictionaries map the original URIs of the entities with the ID of their embeddings.

Gradient Descent Algorithm

Once the numerical representation and dictionaries of the RDF triples are created, the embeddings of the entities can now be *learned*. Learning embeddings happens at the **Encoding Generation** phase. This numerical representation of the data is now fed to the optimization method. The method aims to update the value of the embeddings in order to minimize an overall error according to a proposed model. MateTee is based on TransE, this method aims at minimizing the distance (in MateTee Euclidean Distance is used) between the sum of the embeddings of the Subject and Translation to the embeddings of the Object. TransE also defines *corrupted triples*, which are triples with either the Subject or Object replaced by another randomly selected resource from the set of entities. This is required because TransE needs not only to ensure that similar entities should be close in the embedding space, but also, that dissimilar entities must be farther than the similar ones. This can be seen in the following *Loss Function* used by TransE:

Definition 4 (Loss function) *Given is a set of RDF triples T and their respective set of corrupted RDF triples (original triples with either the Subject or Object replaced) $T^{'}$. Embeddings of Subject s, Object o, and Transitions t in T are represented as $\boldsymbol{S}$, $\boldsymbol{O}$, and $\boldsymbol{T}$, respectively. Similarly, embeddings of Subject $s^{'}$ and Object $o^{'}$ in corrupted RDF triples in $T^{'}$ are represented as $\mathbf{S}'$ and $\mathbf{O}'$, respectively. The loss function can be defined as:*

$$Loss(T,T') = \sum_{(s,t,o)\in T} \sum_{(s^{'},t,o^{'})\in T^{'}} [margin + d(\mathbf{S}+\mathbf{T},\mathbf{O}) - d(\mathbf{S}'+\mathbf{T},\mathbf{O}')]_{+}$$

The key is to notice that the loss function only considers the positive part of the difference of the distances, plus the margin; this is denoted by $[x]_{+}$ in the loss formula. Considering positive values is crucial because if the distance between entities of the original triple, i.e., $d(\mathbf{S}+\mathbf{T},\mathbf{O})$, is greater than the distance between the entities of the corrupted triple, i.e., $d(\mathbf{S}'+\mathbf{T},\mathbf{O}')$, then the difference between the two is positive (regardless of the margin) and this number will increase the overall error. This situation should not occur according to the model $S+T\approx O$ as we want this difference to be as close to zero as possible. On the other hand, if the opposite situation happens, the distance between the entities of the original RDF triple, i.e., $d(\mathbf{S}+\mathbf{T},\mathbf{O})$, is smaller than the distance between the entities of the corrupted triple, i.e., $d(\mathbf{S}'+\mathbf{T},\mathbf{O}')$. This state is exactly what the model looks for, and since the difference between both distances is negative, the overall error is not increased as only the positive part is considered. In the case when the entities of the original RDF triple is smaller than the distance between the entities of the corrupted triple, the margin tightens the model as the negative difference between both distances must be at least as big as the margin, otherwise the overall error will be increased.

TransE - Gradient Descent Algorithm

The core of TransE learning algorithm performs the following steps:

1. **Initialization:** The embedding of each entity (Subject/Object) is initialized uniformly and randomly between $\frac{-6}{\sqrt{k}}$ and $\frac{6}{\sqrt{k}}$ where k is the dimensionality of the embeddings. At this point only the relations are normalized, they will not be normalized again during the optimization. Entities will be normalized at the beginning of each iteration.

2. **Training (loop):**

   a) **Entity embeddings normalization:** In each iteration, first current embeddings of the entities are normalized. This is important because it prevents the optimization to minimize the error by artificially increasing the length i.e., norm, of the embeddings.

   b) **Creation of mini-batches:** Triples to be used as training examples for each iteration of the GD are selected. First, a random sample of the set of triples from the

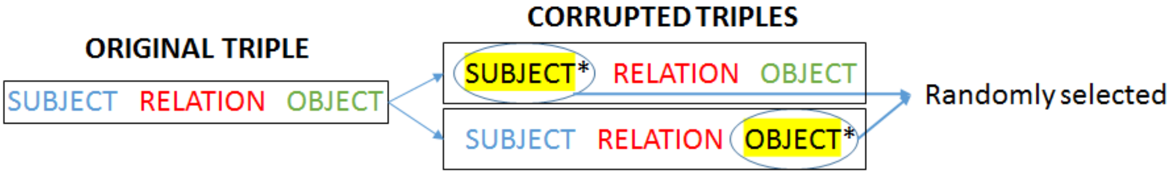


Figure 5.5: **Corrupted triples.** An original RDF triple t and two corrupted versions of t are presented on the left and right hand of the figure, respectively. Corrupted triples have either the Subject or the Object replaced by another randomly selected entity from the input knowledge graph.

input data set is chosen, and then, for each triple in the sample, a corrupted triple is created. A corrupted triple is defined as follows:

- **Corrupted triples:** A corrupted triple is the same as the original but with either its Subject or Object replaced by another randomly selected entity from the data set, always just one, not both at the same time, as show in Figure 5.5:

c) **Embeddings update:** Once the training set of examples, i.e., real triples $\cup$ corrupted triples is set, it proceeds with the actual optimization process:

- For each one of the dimensions of each one of the embeddings in the data set, we calculate the derivative of the overall error with respect to this parameter. This derivative gives the direction on which the overall error is growing with respect to this parameter. Then, to know how to update this parameter so that the overall error decreases, it changes the direction to the opposite of the derivative, and moves one unit of the learning rate (which is also an input hyper-parameter). This process iterates until a maximum number of iterations is reached.

Similarity Mesaure Computation

When the optimization step reaches the termination condition, e.g., the maximum number of iterations in TransE, the embeddings of the entities have been already learned. Having the embeddings of all the entities in the input knolwedge graph, including e_1 and e_2, MateTee can now proceed to the **Similarity Measure Computation** of both entities. Any distance metric for vector spaces can be used to calculate this value, e.g., any Minkowski distance, Euclidean for MateTee. It is important to notice that MateTee calculates the similarity and not the distance. Therefore, using the following Euclidean distance formula, MateTee finds a similarity value between 0 and 1:

$$similarity(A,B) = \frac{1}{1 + EuclideanDistance(A,B)}$$

5.3.3 Empirical Studies

We empirically study the effectiveness of MateTee on measuring the semantic similarity between entities in a knowledge graph. We assess the following research questions:

- **Q1:** Does the translations embeddings method used in MateTee improve the accuracy of determining relatedness between entities in a knowledge graph?

		CESSM 2008	**CESSM 2014**
Size (MBs)		1	1
Triples		8,359	20,153
Entities	Left	1,039	1,559
	Shared	0	0
	Right	1,908	3,909
Relations		1	1

Table 5.5: **CESSM 2008 and 2014 - Dataset description.** Shows dataset size in Megabytes, overall number of triples, number of left entities (Subjects), right entities (Objects), and shared entities (appearing as Subject and as Object), and number of relations, to present a comparison of size between datasets from 2008 and 2014

- **Q2:** Is MateTee able to perform as good as the state-of-the-art similarity metrics?
- **Q3:** Does MateTee perform well in Knowledge Graphs from different domains?

To answer our research questions, we evaluate MateTee in two different scenarios. In the first evaluation, we compare Proteins annotated with the Gene Ontology[5]. In the second evaluation, we compare people extracted from DBpedia, we prepare a dataset named *DBpedia People* [127]. MateTee is implemented in Python 2.7.10. The experiments were executed on a Ubuntu 14.04 (64 bits) machine with CPU: Intel(R) Xeon(R) E5-2660 2.60GHz (20 physical cores) with 132GB RAM, and GPU card GeForce GTX TITAN X. MateTee's source code is available in Git[6].

Similarity among Proteins annotated with the GO ontology

Datasets. This experiment is conducted on the collections of proteins published at the Collaborative Evaluation of GO-based Semantic Similarity Metrics [133] (CESSM) websites 2008[7] and 2014[8]. The CESSM 2008 collection is composed of 13,430 pairs of proteins from UniProt with 1,039 distinct proteins, while the CESSM 2014 dataset includes 22,302 pairs of proteins also from UniProt with 1,559 distinct proteins. The sets of annotations of CESSM 2008 and 2014 comprise 1,908 and 3,909 distinct GO terms, respectively. The original CESSM collections are presented in a multi-file fashion, one file per protein. Technical details in Table 5.5 refer to the unified (single file) dataset, after data transformations are applied. CESSM computes the Pearson's correlation coefficients with respect to three similarity measures from the genomic domain[9]: *ECC similarity* [134], *Pfam* [135], and the *Sequence Similarity* (SeqSim) [136]. Furthermore, the CESSM evaluation framework makes the results of eleven semantic similarity measures available.

These state-of-the-art semantic similarity measures are specific for the genomic domain and exploit the knowledge encoded in the Gene Ontology (GO) to determining relatedness among proteins in the CESSM collections. These semantic similarity measures are extensions of well-known similarity measures to consider GO annotations, Information Content (IC) of these annotations, and pair-wise combinations of common ancestors in GO hierarchy. The extended

[5] http://geneontology.org/

[6] https://github.com/RDF-Molecules/MateTee

[7] http://xldb.di.fc.ul.pt/tools/cessm/

[8] http://xldb.di.fc.ul.pt/biotools/cessm2014/

[9] The area in molecular biology and genetics that studies the genetic material of an organism.

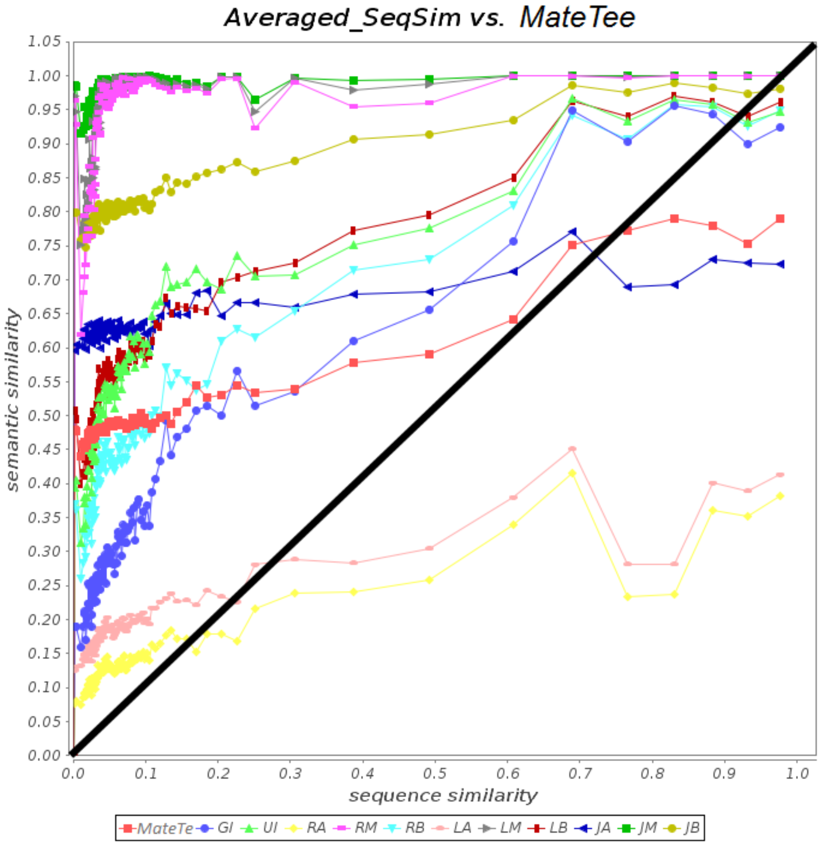


Figure 5.6: **Results from the CESSM evaluation framework for the CESSM 2008 collection.** Results include: average values for MateTee with respect to SeqSim. The black diagonal line represents the values of SeqSim for the different pairs of proteins in the collection. The similarity measures are: simUI (UI), simGIC (GI), Resnik's Average (RA), Resnik's Maximum (RM), Resnik's Best-Match Average (RB/RG), Lin's Average (LA), Lin's Maximum (LM), Lin's Best-Match Average (LB), Jiang & Conrath's Average (JA), Jiang & Conrath's Maximum (JM), J. & C.'s Best-Match Average. (JB). MateTee outperforms eleven measures and reaches a value of Pearson's correlation of **0.787**.

similarity measures are the following: Resnik (R) [137]; Lin (L) [138]; and Jiang and Conrath (J) [139]. Additionally, the CESSM evaluation framework considers the average of the ICs of pairs of common ancestors during the computation of these measures; this measure is denoted with the label *A*. Following the approach reported by Sevilla et al. [140], the maximum value of IC of pairs of common ancestors is computed; combined measures are distinguished with the label *M*. As proposed by Couto et al. [141], the best-match average of the ICs of pairs of disjunctive common ancestors (DCA) is also computed; measures labelled with *B* or *G* correspond to combinations with the best-match average of the ICs. Finally, the Jaccard index is applied to sets of annotations together with domain-specific information in the similarity measures simUI (UI) and simGIC (GI) [142].

Results. Figures 5.6 and 5.7 report on the comparison of MateTee and the rest of the eleven similarity measures with SeqSim; both plots were generated by the CESSM evaluation framework. The black diagonal lines represent the values assigned by SeqSim. The majority of the studied similarity measures assign high values of similarity to pairs of proteins that SeqSim

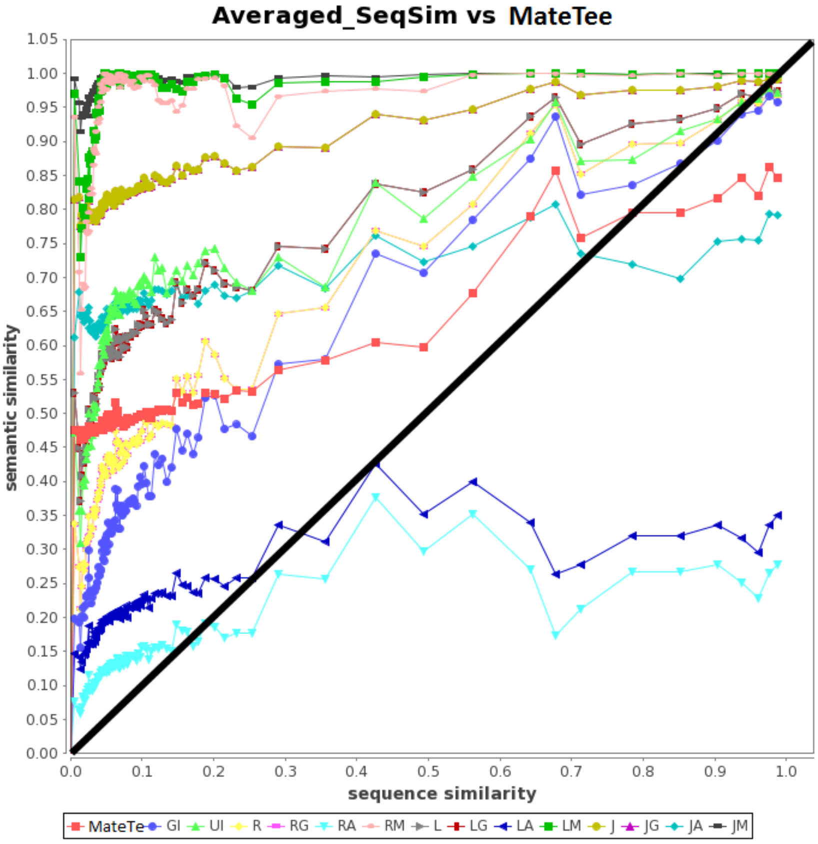


Figure 5.7: **Results from CESSM evaluation framework for the CESSM 2014 collection.** Results include: average values for MateTee with respect to SeqSim. The black diagonal line represents the values of SeqSim for the different pairs of proteins in the collection. The similarity measures are: simUI (UI), simGIC (GI), Resnik's Average (RA), Resnik's Maximum (RM), Resnik's Best-Match Average (RB/RG), Lin's Average (LA), Lin's Maximum (LM), Lin's Best-Match Average (LB), Jiang & Conrath's Average (JA), Jiang & Conrath's Maximum (JM), J. & C.'s Best-Match Average. (JB). MateTee outperforms eleven measures and reaches a value of Pearson's correlation of **0.817**.

considers as similar proteins, i.e., in pairs of proteins with high values of SeqSim, the majority of the curves of the similarity measures are close to the black line. Nevertheless, the same behavior is not observed for the pairs of proteins that are not similar according to SeqSim, i.e., the corresponding curves are far from the black line. Contrary to state-of-the-art similarity measures, MateTee is able to compute values of similarity that are more correlated to SeqSim, i.e., the curve of MateTee is close to the black line in both collections. MateTee is able to reach values of the Pearson's correlation of **0.787** and **0.817** in CESSM 2008 and 2014, respectively.

Additionally, we present the comparison of MateTee and eleven similarity measures with respect to the gold standard similarity measures: ECC, Pfam, and SeqSim; Table 5.6 presents the results, including five additional similarity measures, i.e., d_{tax} [131], d_{ps} [143], OnSim [144], IC-OnSim [145], and GADES [146]. As before, values of the Pearson's correlation represent the quality of a measurement of similarity, the higher the correlation with the gold standards, the better the measurement. The top 5 similarity measures (before introducing MateTee) with higher quality are highlighted in gray, and the highest is highlighted in bold.

Discussion: From the results, the following insights can be concluded; MateTee already outperforms the quality of GADES for both collections 2008 and 2014, which is the best-performing measurement before our method, for the Sequence Similarity. In the 2008 collection, MateTee stands at the 5th position against the other two gold standards, only at 0.015 points to the GADES for ECC, and 0.043 for Pfam. While in the 2014 collection, MateTee stands at the 3th position against the Pfam gold standard, only at 0.029 points to the GADES (the best before MateTee), and at the 5th position against the ECC gold standard, only at 0.014 points to the GADES (the best before MateTee).

It can be observed that GADES [146] is the greatest competitor for MateTee. It performs better comparing with the ECC and Pfam gold standards, but it is outperformed against SeqSim. As the results of GADES and MateTee are rather close. For the three gold standards, the advantage of MateTee against GADES is that the former requires domain expertise to define its final similarity measure (GADES defines multiple measures based on: Class hierarchy, Neighborhood, Relation Hierarchy, Attributes, and mixtures of them). While the latter learns the embeddings in an automatic way (through an optimization process called Stochastic Gradient Descent), and then uses any common vector similarity measure, e.g., Euclidean or Cosine, to calculate their similarity.

Similarity		2008			2014	
measure	*SeqSim*	*ECC*	*Pfam*	*SeqSim*	*ECC*	*Pfam*
GI [142]	0.773	0.398	0.454	0.799	0.458	0.421
UI [142]	0.730	0.402	0.450	0.776	0.470	0.436
RA [147]	0.406	0.302	0.323	0.411	0.308	0.264
RM [148]	0.302	0.307	0.262	0.448	0.436	0.297
RB [149]	0.739	0.444	0.458	0.794	0.513	0.424
LA [150]	0.340	0.304	0.286	0.446	0.325	0.263
LM [148]	0.254	0.313	0.206	0.350	0.460	0.252
LB [149]	0.636	0.435	0.372	0.715	0.511	0.364
JA [151]	0.216	0.193	0.173	0.517	0.268	0.261
JM [148]	0.234	0.251	0.164	0.342	0.390	0.214
JB [149]	0.586	0.370	0.331	0.715	0.451	0.355
d_{tax} [131]	0.650	0.388	0.459	0.682	0.434	0.407
d_{ps} [143]	0.714	0.424	0.502	0.750	0.480	0.450
OnSim [144]	0.733	0.378	0.514	0.774	0.455	0.457
IC-OnSim [145]	0.779	0.443	**0.539**	0.810	0.513	0.489
GADES [146]	0.780	**0.446**	**0.539**	0.812	**0.515**	**0.49**
MateTee	**0.787**	0.431	0.496	**0.817**	0.501	0.461

Table 5.6: **GO - CESSM 2008 and 2014 - Results.** Quality in terms of Pearson's correlation coefficient between three gold standards, i.e, SeqSim (Sequence) similarity, Pfam (Protein Families) similarity and EC (Enzyme Commission) similarity, and thirteen in-house similarity measures of CESSM, plus OnSim, IC-OnSim and GADES. With gray background the best 6 correlations and the best in bold.

		Data set 1	Data set 2	Data set 3
Size (MBs)		80	80	80
Triples		552,355	553,232	552,527
Subjects (Persons)		20,000	20,000	20,000
Entities	Left	60,000		
	Shared	0		
	Right	247,465		
Relations		1,981		

Table 5.7: **Description of the data set *DBpedia People*.** Shows the datasets size in Megabytes, overall number of triples, overall number of persons, number of left entities (Subjects), right entities (Objects), and shared entities (appearing as Subject and as Object), and number of relations, to present a comparison of size between three dumps

Similarity among People from DBpedia

Dataset: Table 5.7 shows technical details of the datasets used in the *DBpedia People* experiment. The **Gold Standard (GS)** was extracted from the live version of DBpedia (July 2016). It contains 20,000 subjects of type Person[10], i.e., 20,000 subjects with all available properties and their values. The overall number of RDF triples is 829,184. The Gold Standard is used to compute Precision and Recall during the evaluation. The **Test Datasets (TS)** are created from the Gold Standard with their properties and values were randomly split among three test datasets. Each triple is randomly assigned to one or several test datasets. The selection process takes two steps: 1) a number of test datasets to copy a triple to is chosen randomly under a uniform distribution; 2) the chosen number is used as a sample size to randomly select particular test datasets to write a triple. URIs are generated specifically for each test data set. Eventually, each test dump contains a randomly uniform subset of the properties in the gold standard.

Metrics: We measure the behavior of MateTee in terms of the following metrics:

- **Precision** From all matched pairs (pairs with similarity greater than the threshold), percentage of correct matches.

$$Precision = \frac{\text{Number of correctly matched pairs}}{\text{Total number of matched pairs}}$$

- **Recall** From all expected matches (all, including below and above the threshold), percentage of correct matches.

$$Recall = \frac{\text{Number of correctly matched pairs}}{\text{Total number of expected matches}}$$

Results: We tested the quality of MateTee by comparing its results with two other similarity measurements: Jaccard (Section 5.2.1) and GADES (Section 5.2.2). For each one we calculate the Precision and Recall, considering different values of **Threshold**. The Threshold is the minimum similarity value so that the pair of people is considered in the matched-pairs set. Table 5.8 show the results obtained using Jaccard, GADES, and MateTee similarity approaches.

[10] <http://dbpedia.org/ontology/Person>

	T0.6		**T0.7**		**T0.8**		**T0.9**	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
Jaccard	0.36	0.01	0.30	0.01	0.30	0.01	0.30	0.01
GADES	0.87	0.73	0.83	0.43	0.80	**0.16**	**0.63**	**0.05**
MateTee	**0.93**	**0.79**	**0.99**	**0.59**	**1.00**	0.10	0.00	0.00

Table 5.8: **DBpedia People Test Datasets.** Results comparison of precision and recall using Jaccard, GADES and MateTee similarity measurements, obtained with different threshold values: 0.6, 0.7, 0.8 and 0.9. In bold the best value for each threshold

Discussions: From the results we extract the following insights. Regarding Precision, MateTee similarity measurement has the best quality among all the three measurements, and for all the considered thresholds. Regarding Recall, our method is the best up until a threshold of 0.7. For higher thresholds, e.g., 0.8, the recall rapidly goes down to 0.1, and to absolute 0 for 0.9. The explanation for this is that MateTee, being an optimization-based method, will always have an error as small as possible, so even if the neighborhoods of two entities are exactly the same, it is very unlikely to have similarities higher than 0.9 or 1.0, they will for sure be higher than between people which neighborhoods are absolutely different, but very unlikely be equal to 1.0. Then, using a threshold equal to 0.9, very few pairs of people will be considered, and with 1.0, absolutely no pairs are considered to count in the numerator of the Recall formula.

5.4 Summary

A similarity metric is a key building in the integration process of semantic equivalent entities from web sources. In this chapter, we studied the effectiveness of GADES (a semantic similarity metric) versus Jaccard (a non-semantic similarity metric). To do so, we adapted both GADES and Jaccard to work with RDF molecules. The observed results suggest that GADES performs better since it uses the semantics encoded in the RDF molecules. We presented as well MateTee, a method to compare entities in knowledge graphs, based on the vector representation of the entities (embeddings) created automatically without any domain expertise. We compared MateTee versus several state-of-the-art methods including GADES, OnSim, and metrics available in the CESSM evaluation framework. MateTee exhibited competitive results, even outperforming GADES' results, one of the best-performing similarity metric. GADES and MateTee are both semantic similarity metrics, they take advantages of the semantics encoded in the RDF molecules, e.g., classes and relationships. Both show good performance on the task of determining the relatedness among RDF molecules and they can be plugged into MINTE integration pipeline.

CHAPTER 6

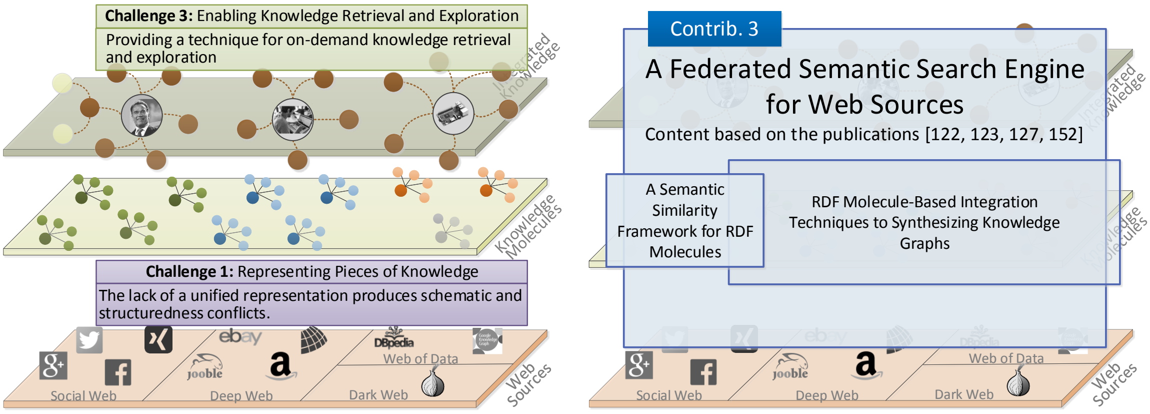
On-Demand Knowledge Retrieval and Exploration Engine for the Web

Heterogeneous web sources contain knowledge about the same entity. To build a complete knowledge graph, we need to collect and integrate the knowledge about entities spread over web sources. Of equal importance, we need to facilitate the exploration of the resulting knowledge graphs. In this chapter, we focus on the problem of building and exploring knowledge graphs on-demand from heterogeneous web sources, the content of this chapter is based on the publications [122, 123, 127, 152]. This chapter answers the following research question:

> **RQ3**: How can knowledge graphs be populated on-demand with data collected from heterogeneous web sources?

We start the chapter by presenting the problems of knowledge retrieval and exploration over web sources in Section 6.1. Particularly, we analyze the problem of on-demand knowledge retrieval and exploration, which is important in the scope of this thesis. Then, in Section 6.2 we present our solution to the research question **RQ3**, i.e., a federated semantic search engine named FuhSen. FuhSen is a keyword-based federated engine that exploits the search capabilities of heterogeneous sources during query processing and generates knowledge graphs on-demand applying an RDF molecule integration approach in response to keyword-based queries. The resulting knowledge graph describes the semantics of entities collected from the integrated sources, as well as relationships among these entities. At first, in Section 6.2.1, we formalize the problem that FuhSen is solving. FuhSen's core relies on the integration approach MINTE (cf. Chapter 4) and in the semantic similarity framework (cf. Chapter 5). After, we explain the three main steps performed by FuhSen, i.e., the creation of the RDF molecules in Section 6.2.2, the integration of these RDF molecules of data in Section 6.2.3, and finally the exploration of the synthesized RDF graphs in Section 6.2.4.

Furthermore, we conducted empirical evaluations where FuhSen is compared to traditional search engines. FuhSen semantic search capabilities, supported by domain ontologies, allow users to complete search tasks that could not be accomplished with traditional Web search engines during the evaluation study. Section 6.3 presents the results of our empirical studies on FuhSen. The evaluation results suggest that FuhSen is able to accurately integrate data in a knowledge graph than from heterogeneous web sources. Finally, the closing remarks of this chapter are pointed out in Section 6.4. In summary, the contributions of this chapter are:


Challenge 3: Enabling Knowledge Retrieval and Exploration
Providing a technique for on-demand knowledge retrieval and exploration
Integrated Knowledge
Knowledge Molecules
Challenge 1: Representing Pieces of Knowledge
The lack of a unified representation produces schematic and structuredness conflicts.
Social Web
Deep Web
Dark Web
Web of Data
Web Sources
Contrib. 3
A Federated Semantic Search Engine for Web Sources
Content based on the publications [122, 123, 127, 152]
A Semantic Similarity Framework for RDF Molecules
RDF Molecule-Based Integration Techniques to Synthesizing Knowledge Graphs
Integrated Knowledge
Knowledge Molecules
Social Web
Deep Web
Dark Web
Web of Data
Web Sources

(a) Problems tackled in this chapter

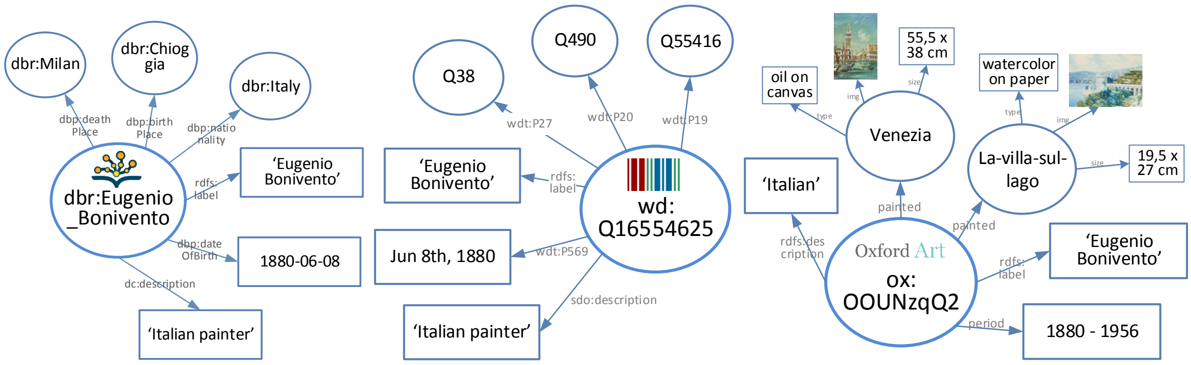
(b) Contributions described in this chapter

Figure 6.1: **Challenges and Contributions:** This chapter focuses on the problem of retrieving and integrating pieces of knowledge from web sources and proposes a federated semantic search engine to build knowledge graphs on-demand.

- A federated hybrid search concept over highly heterogeneous data sources using a semantic aggregation of the distributed information in its core. To the best of our knowledge, this is the first approach targeting such a diverse set of data modalities in a federated manner with semantic aggregation.
- A component-based architecture, where every element can evolve and be replaced with an improved version without affecting the others as well as a comprehensive open-source implementation of the architecture.
- A reactive component-based UI approach that handles the uncertainty imposed by the intrinsic nature of RDF graphs. Additionally, we present its open source proof-of-concept implementation using modern Web UI development technologies.

6.1 The Problem of Knowledge Retrieval and Exploration

The strong support that Web based technologies have received from researchers, developers, and practitioners has resulted in the publication of data from almost any domain. Additionally, standards and technologies have been defined to query, search, and manage Web accessible data sources. A vast amount of information about various types of entities is spread over several parts of the Web, e.g., people or organizations on the Social Web, product offers on the Deep Web or on the Dark Web. These data sources can comprise heterogeneous data and are equipped with different search capabilities, e.g., the Google+ API can return the profile of a user, while the Twitter API also allows for finding the trends of a place. For example, Web access interfaces or APIs allow for querying and searching sources like DBpedia, Wikidata, or the Oxford Art archive. Web sources make overlapping as well as complementary data available about entities, e.g., people, organizations, or art paintings. However, these entities may be described in terms of different vocabularies by these web sources, and data that correspond to the same real-world entities then needs to be integrated in order to have a more complete description of these entities. End users such as investigators from law enforcement institutions searching for traces and connections of organized crime have to deal with these interoperability problems not only



(a) DBpedia data (b) Wikidata data (c) Oxford Art data

Figure 6.2: **Motivating Example of Knowledge Retrieval**. *Eugenio Bonivento* on different web sources is represented as RDF molecules.

during search time (Knowledge Retrieval) but also while exploring the collected information from different sources (Knowledge Exploration).

6.1.1 On-Demand Knowledge Retrieval Challenges

In the crime investigation process, collecting and analyzing information from different sources is a key step performed by investigators. Although scene analysis is always required, a crime investigation process can greatly benefit from searching information about people and products on the Web. Consider a case of counterfeit paintings of *Eugenio Bonivento*, investigators need to gather all the information about the painter and his work. General domain knowledge bases such as DBpedia or Wikidata (Web of Data) contain common information about *Eugenio Bonivento*, while domain-specific web sources like the Oxford Art archive (Deep Web) contain detailed information about his paintings. Figure 6.2 illustrates the RDF molecules of *Eugenio Bonivento* present in these different web sources. DBpedia and Wikidata RDF molecules can be integrated to produce a complete profile of *Eugenio Bonivento*, while Oxford Art completes the paintings information. However, there are heterogeneity problems at the schema and data levels. Each data source provides RDF molecules described in its own vocabulary (schema conflicts) and the same fact might be expressed differently (data conflicts), e.g., the dates in Figure 6.2. Current data integration approaches are performed by experts and it is extremely cumbersome and time-consuming as it requires to access a large number of different data sources and set up a whole integration infrastructure as in [76]. To facilitate the integration of the data about *Eugenio Bonivento*, similarity measures able to decide on the relatedness of the corresponding RDF molecules, and equivalent or complementary properties are required. The following are the more relevant problems we need to solve in an on-demand knowledge retrieval scenario:

P1. **Heterogeneous data sources.** This refers to the ability to search in multiple and heterogeneous web sources. The platform should hide the complexity of data search, extraction and homogenization. The high degree of heterogeneity is defined in terms of data formats, structures, coverage, size, and accessibility.

P2. **Extensible by design.** This means being able to add or remove sources of information, in the platform. All data sources defined for the platform are dynamic by nature. On the *Social Web*, a new social network may gain relevance or expose a new version of its API. On the *Deep Web*, a new e-commerce platform can hit the marketplace and become a valuable source of information. Finally, the *Data Web* is growing very fast, with new valuable open dataset appearing continuously.

P3. **No index creation.** Our approach is neither crawling nor mining the different data sources defined in the vision of the platform. This is related with privacy issues and copyrights to index content of the data sources, especially when they contain sensitive personal data. Instead, our platform should be able to search in real-time information about entities in its data sources.

P4. **Efficiency and Findability.** The speed of processing queries from a client should be as fast as possible. As the platform deals with Big Data sources, such as Social Networks and the Data Web, the integration process should be designed for an acceptable performance from the beginning. As described in [10], retrieving everything that is relevant to the user is the most important requirement for any search engine regardless of its type. So the platform should avoid making the user review irrelevant content.

P5. **Provenance.** With information coming from different sources, it is critical to maintain its provenance. The goal is to track the origin of every piece of information. This is relevant in the domain of *criminal investigation* because investigators must decide whether to accept some piece of information as valid or to carry out further steps, such as an in-place verification of the information.

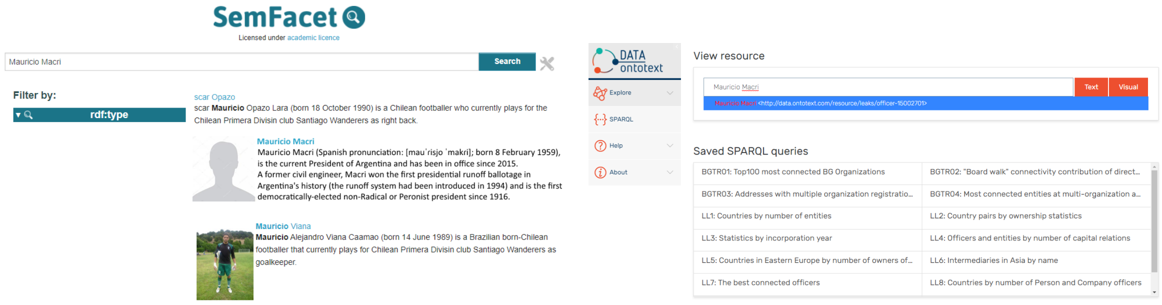
6.1.2 On-Demand Knowledge Exploration Challenges

Let us assume the following *distributed web source* browsing scenario in the context of a crime investigation: during an ongoing investigation for corruption, browsing and analyzing information coming from various sources is one of the key steps performed by investigators. In the case of a "politically exposed person", such as a politician, an investigator wants to explore whether or not this politician is in any form involved or related to the Panama Papers scandal, and at the same time retrieve additional general information about the politician. While the Linked Leaks Dataset[1] contains an RDF representation of the Panama Papers, DBpedia[2] contains general information about politicians. Figure 6.3 illustrates how a user typically requires two different UIs for exploring information about *Mauricio Macri* in two RDF graphs. In this case, SemFacet (Figure 6.3(a)) is used to browse the RDF graph of DBpedia, while the OntoText browser (Figure 6.3(b)) is used to explore the RDF graph of Linked Leaks.

Additionally, the exploration of on-demand built knowledge graph brings new challenges at the UI level. The UI has to deal with a higher degree of uncertainty, such as connectivity problems and longer query response times. These are issues which are only aggravated by the variations in the size of the retrieved data, and the complexity in the semantics of the data, all factors which potentially have a negative effect on the usability of the interface, ultimately progressing into a decrease of the overall user experience. State-of-the-art approaches [113, 115, 116, 118] are mainly designed to explore one RDF graph at a time. Consequently, they do not address these challenges sufficiently. In the following, we will briefly describe what we identified as some of the main UI problems when browsing on-demand built knowledge graphs.

[1] http://data.ontotext.com/linkedleaks

[2] http://wiki.dbpedia.org



scar Opazo
scar **Mauricio** Opazo Lara (born 18 October 1990) is a Chilean footballer who currently plays for the Chilean Primera Divisin club Santiago Wanderers as right back.

Mauricio Macri
Mauricio Macri (Spanish pronunciation: [mauˈrisjo ˈmakri]; born 8 February 1959), is the current President of Argentina and has been in office since 2015.
A former civil engineer, Macri won the first presidential runoff ballotage in Argentina's history (the runoff system had been introduced in 1994) and is the first democratically-elected non-Radical or Peronist president since 1916.

Mauricio Viana
Mauricio Alejandro Viana Caamao (born 14 June 1989) is a Brazilian born-Chilean footballer that currently plays for Chilean Primera Divisin club Santiago Wanderers as goalkeeper.

Mauricio Macri <http://data.ontotext.com/resource/leaks/officer-15002701>

BGTR01: Top100 most connected BG Organizations	BGTR02: "Board walk" connectivity contribution of direct...
BGTR03: Addresses with multiple organization registratio...	BGTR04: Most connected entities at multi-organization a...
LL1: Countries by number of entities	LL2: Country pairs by ownership statistics
LL3: Statistics by incorporation year	LL4: Officers and entities by number of capital relations
LL5: Countries in Eastern Europe by number of owners of...	LL6: Intermediaries in Asia by name
LL7: The best connected officers	LL8: Countries by number of Person and Company officers

(a) Exploring DBpedia using SemFacet (b) Exploring Linked Leaks using OntoText's UI

Figure 6.3: **Motivating Example of Knowledge Exploration.** A user typically requires two different UIs to explore the RDF graphs of DBpedia and Linked Leaks.

P1. **Reactiveness towards the semantics of the data** The quality and amount of the semantics of data in a *on-demand built knowledge graph exploration* scenario vary in a significant way. We may find general concepts (e.g., Organization) to more specialized concepts (e.g, Terrorist Organization). The UI needs to deal and react to the variety of the semantics in the data, in the sense of preserving the semantics of the information being retrieved. Most of the current implementations [35, 116] are designed with single RDF knowledge graph exploration scenarios in mind, where flows of information in the form of entities and attributes that originate from them are static in structure. In such cases, designing interfaces that implement **reactivity** in the semantic context have received more attention of researchers recently. However, this is unfortunately not the case when it comes to the exploration of on-demand built knowledge graphs. This is due to the fact that the flow of information in such cases is no longer static in structure – it is inherently different, constantly growing and evolving, thus posing a challenge when it comes to designing standardized but scalable interfaces that cope with such change. Hence, we identified a lack of reactive approaches that tackle semantic contexts when it comes to on-demand built knowledge graph exploration. We tackle this problem by developing a reactive UI components that allows on-demand built knowledge graph exploration while preserving the semantic contextualization of the results.

P2. **Error visibility and feedback** Another potential pitfall that can be observed when *exploring on-demand built knowledge graphs* is that of error visibility and feedback. Errors during the browsing process need to be made clearly visible so as to provide better feedback to the users, in accordance to the *visibility of the system status* and the *help users recognize, diagnose, and recover from errors* heuristics referred to by Nielsen [153]. At the same time, the attention span time frames mentioned by Nielsen [154] should also be preserved. These usability heuristics help safeguard that interfaces are sufficiently comfortable and functional when the user performs the intended tasks of a system, ensuring further adoption of the technology. However, these principles are difficult to preserve in the context of on-demand built knowledge graph exploration, due to the nature of how federated searches operate—different queries for different knowledge graph sources, performed in parallel, typically imply different response times for the queries as well. This contributes to longer waiting times, where the user could be left without any feedback for time spans that surpass a few seconds, causing confusion and triggering a sense of lack of control in the users, which ultimately breaks the feedback flow between the interface and the user. We attempt to tackle this problem by the implementation of a reactive UI component, which allows for improved visibility and feedback of errors that may happen during the federated search process, where the user can have full feedback as to the current system status, as well as obtaining information about the nature of the errors that have occurred.

P3. **Minimalist design (P3)** A UI for exploring RDF graphs needs to manage well the topic of *minimalism* and avoid violating the *aesthetics and minimalist design* rules that also form part of Nielsen's heuristics [153]. To address these rules, we took three main requirements into consideration when designing our approach: *maximum screen space usability*, *maximum visibility* (to also contribute in solving **P2**), and *minimal cluttering.* However, providing *minimal cluttering* in a screen space where there is both abundant information to display and considerable functionality to support becomes a challenge, especially in systems which have multi-layer navigation bars, as they either consume an important block of screen space or become too confusing or difficult to follow for the user. Thus, these design requirements were also carefully considered when designing our UI components. Moreover, the implementation of reactive features in our design further aims to contribute to minimizing the required screen space, as well as improving the ergonomics of the interface by providing familiarity in the interface by means of semantically enabling or disabling views or functions according to the results being obtained during the search.

6.2 A Federated Semantic Search Engine

In this section, we present FuhSen a novel federated semantic search engine. FuhSen exploits state-of-the-art semantic similarity measures, and integrate properties on-demand of any type of entity e.g., a person *Eugenio Bonivento* into a single RDF knowledge graph from web sources.

6.2.1 Problem Definition

Our federated semantic search engine *FuhSen* addresses the challenges described in Section 6.1. In this section, we formally define the problem as follows. Given a keyword query, i.e., a set of strings containing one or more entities. *FuhSen* creates a knowledge graph at query time that represents the entities associated with the keywords in the query, and utilizes semantic similarity measures to determine the relatedness of entities to be integrated. A knowledge graph is composed of a set of entities, their properties, and relations among these entities. The Semantic Web technology stack provides the pieces required to define and build a knowledge graph. To properly understand these concepts, we follow the notation proposed by Arenas et. al. [27], Piro et. al. [155], and Fernandez et. al. [8], Ribón et. al. [7] to define RDF triples, knowledge graphs, RDF molecules, and similarity measures, respectively.

Definition 5 (RDF triple [27]) *Let* $\mathbf{I}$, $\mathbf{B}$, $\mathbf{L}$ *be disjoint sets of URIs, blank nodes, and literals, respectively. A tuple* $(s, p, o) \in (\mathbf{I} \cup \mathbf{B}) \times \mathbf{I} \times (\mathbf{I} \cup \mathbf{B} \cup \mathbf{L})$ *is denominated an RDF triple, where* s *is called the subject,* p *the predicate, and* o *the object.*

Definition 6 (Knowledge Graph [155]) *Given a set* T *of RDF triples, a knowledge graph is a pair* $G = (V, E)$*, where* $V = \{s \mid (s, p, o) \in T\} \cup \{o \mid (s, p, o) \in T\}$ *and* $E = \{(s, p, o) \in T\}$.

Definition 7 (RDF Subject Molecule) *Using the defintion we provided in Section 4.2, we define an RDF molecule as follows: Given an RDF graph G, we call a subgraph M of G an RDF molecule iff the RDF triples of* $M = \{t_1, \ldots, t_n\}$ *share the same subject, i.e.,* $\forall\ i, j \in \{1, \ldots, n\}$ $(subject(t_i) = subject(t_j))$. *An RDF molecule can be represented as a tuple* $\mathcal{M} = (R, T)$*, where R corresponds to the URI (or blank node ID) of the molecule's subject, and T is a set of pairs* $p = (prop, val)$ *such that the triple* $(R, prop, val)$ *belongs to M. Property values are free of blank nodes, i.e., let I be a set of IRIs and L a set of literals, then* $val \in I \cup L$.

Definition 8 (Individual similarity measure [7]) *Given a knowledge graph* $G = (V, E)$*, two entities* e_1 *and* e_2 *in V, and a resource characteristic RC of* e_1 *and* e_2 *in G, an individual similarity measure* $Sim_{RC}(e_1, e_2)$ *corresponds to a similarity function defined in terms of RC for* e_1 *and* e_2.

Definition 9 (Aggregated similarity measure [7]) *Given a knowledge graph* $G = (V, E)$ *and two entities* e_1 *and* e_2 *in V, an aggregated similarity measure* α *for* e_1 *and* e_2 *is defined as* $\alpha(e_1, e_2 \mid T, \beta, \gamma) := T(\beta(e_1, e_2), \gamma(e_1, e_2))$ *where:*

- *T is a triangular norm (T-Norm) [156].*
- $(\beta(e_1, e_2)$ *and* $\gamma(e_1, e_2))$ *are aggregated or individual similarity measures.*

FuhSen leverage semantic similarity measures to address a research problem: given a keyword query *Q*, a threshold *T*, build a knowledge graph of heterogeneous data which are no less semantically similar than *T*. Figure 6.2 presents three RDF molecules with data about *Eugenio Bonivento* collected from DBpedia, Wikidata, and Oxfort Art, respectively. Each of the data sources applies its own approach for knowledge serialization, e.g., DBpedia employs human-readable URIs whereas Wikidata encodes entities with auto-generated identifiers as combinations of letters and numbers which is hard to comprehend without prior acquaintance with the Wikidata data model. Evidently, simple string similarity metrics will fail to identify a possible link among those molecules due to a lack of shared common string literals. Semantics of the facts encoded in RDF molecules has to be considered in order to truly grasp their similarity. In other words, a new, higher abstraction layer has to be established. Such a level, which operates on semantic knowledge instead of symbols (in which the knowledge is presented), allows for semantic similarity measures. The following section introduces and describes the architecture of *FuhSen*, a system that is capable of exploiting the MINTE framework (cf. Chapter 4), and solving the knowledge retrieval and exploration problem described in Section 6.2.1.

6.2.2 Creation of RDF Molecules

As an input, *FuhSen* receives a keyword query *Q*, e.g., *Eugenio Bonivento*, a similarity metric, a fusion policy, and threshold value *T*, e.g., 0.7. The input values are processed by the *Query Rewriting* module, which formulates a correct query to be sent to the *Mediator-Wrapper* component. The *Mediator* explores all RDF Wrappers in the federation and using the Definition 7 transforms the output into RDF molecules under the OntoFuhSen vocabulary. Intermediate results are enriched with additional knowledge in the *RDF Molecules Enrichment* module. Finally, molecules with materialized induced facts are integrated into a knowledge graph in the *RDF Molecule Integration* module. The integration module uses the MINTE Framework described in Chapter 4, and it consists of three sub-modules responsible for: 1) identifying

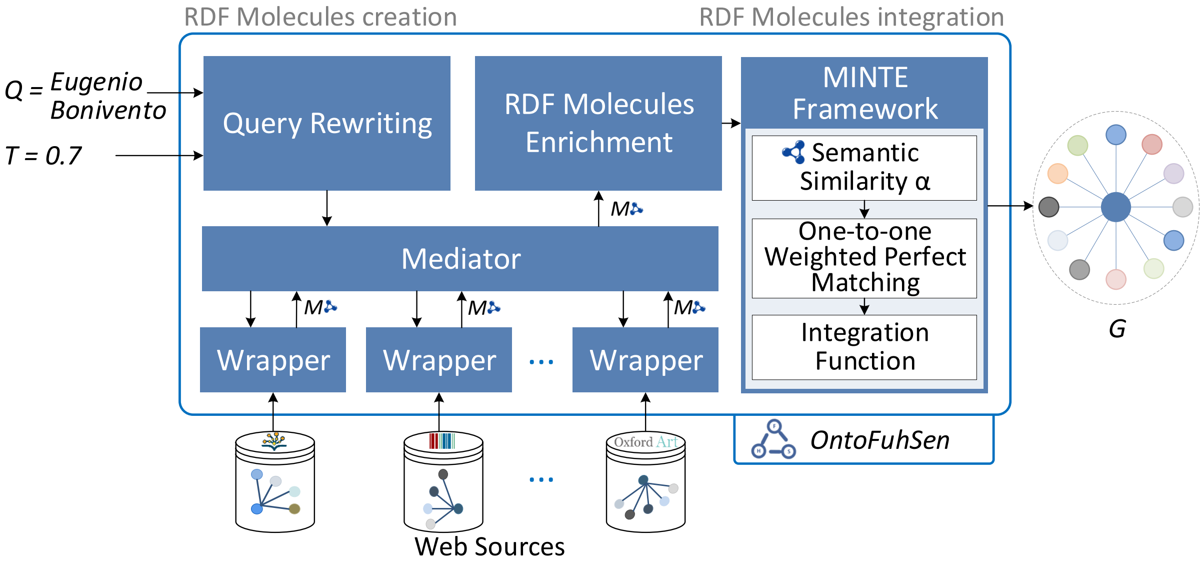


Figure 6.4: **The *FuhSen* Architecture**. *FuhSen* receives a keyword query Q and a threshold T, and produces a knowledge graph G populated with the entities associated with the keywords in the query and their relationships. Input queries are rewritten into queries understandable by the available data sources. Wrappers are used to collected the data from the relevant sources and to create RDF molecules. Values of semantic similarity measures are computed pair-wise among RDF molecules, and the 1-1 weighted perfect matching is computed to the determine the most similar RDF molecules. RDF molecules connected by an edge in the solution of the 1-1 weighted perfect matching are merged into a single RDF molecule in knowledge graph G.

semantic similarity of molecules; 2) performing one-to-one perfect matching; and 3) integrating similar RDF molecules. Figure 6.4 shows the main modules of the FuhSen approach. We describe each component in detail.

The OntoFuhSen Vocabulary

The *OntoFuhSen*[3] vocabulary serves as a global schema for the federated search engine to retrieve and integrate data coming from different web sources. The *OntoFuhSen* vocabulary allows for describing the data sources, and entities in the federation. The rationale of the vocabulary is threefold: *1*) facilitating visualization and faceted browsing of the results; *2*) acting as a unified data schema on top of which semantic algorithms can enhance the completeness of search results; and *3*) as a response format for exchanging data collected from the wrappers with the rest of the FuhSen's engine components. Additionally, the *OntoFuhSen* vocabulary allows for the description of user search activities, data sources, and entities in the federation (cf. Figure 6.5). The vocabulary is divided into the following three modules:

(1) Search engine metadata: comprises classes modeling a user's search activity (e.g., `fs:Search`, `fs:SearchableEntity`). This module takes into account the provenance of resources. To enable provenance tracking, classes of the *PROV*[4] standard vocabulary have been extended to model the provenance of the information related to a user's search activities.

(2) Data source metadata: contains classes describing Web API services and access points

[3] https://w3id.org/eis/vocabs/fuhsen
[4] http://www.w3.org/ns/prov

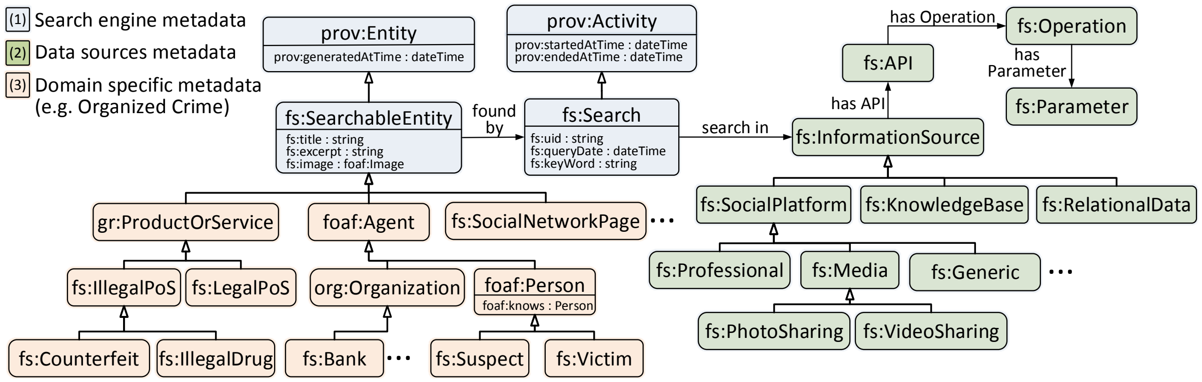


Figure 6.5: **An Overview of the OntoFuhSen vocabulary.** The three modules of the OntoFuhSen vocabulary are depicted in different colors; main classes of each module are presented.

(e.g., `fs:Parameter`, `fs:Operation`). They model data sources from which the RDF molecules are collected, e.g., Facebook, DBpedia, Twitter, or Google Knowledge Graph.

(3) Domain specific metadata: includes classes for describing the results collected from *FuhSen* during keyword query processing. For the crime domain concepts include: `gr:ProductOrService` and `org:Organization`. The *FuhSen* vocabulary utilizes existing well-known ontologies, e.g., terms from *FOAF* and *Schema.org* [5].

Query Rewriting

This component basically transforms the initial keyword query to queries that the wrappers understand. Using the data source description in the OntoFuhSen vocabulary the initial query is transformed into, e.g., a SPARQL query or a REST API request depending on the case. The final list of queries is sent to the search engine component.

Wrapper-Mediator Components

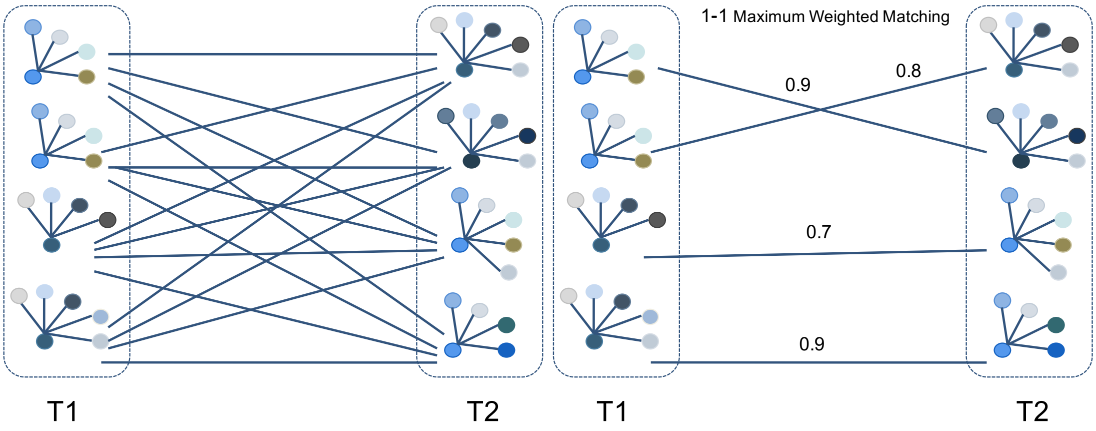
The wrapper-mediator components orchestrate the data extraction process using RDF wrappers and store the RDF molecules in an in-memory graph. The search engine receives the keyword query and, based on the data sources' description defined in terms of the OntoFuhSen vocabulary, orchestrates in an asynchronous manner the RDF molecules creation. Requests to the RDF wrappers are created based on the Web APIs[6] of the data sources, whose wrappers are described in terms of *OntoFuhSen*. Once a result has been received from a wrapper, a request to aggregate it in the results knowledge graph is sent to the vocabulary-based aggregator component. The aggregator creates an in-memory *RDF* graphs containing the RDF molecules, where all responses produced by the RDF wrappers are aggregated and described using *OntoFuhSen*. The vocabulary-based approach keeps the data aggregation task relatively simple.

RDF Molecules Enrichment

Once the RDF molecules have been constructed, *FuhSen* allows for additional quality improvement by enriching them with new facts acquired through the *typing* process [157]. It is thus

[5] `http://xmlns.com/foaf/spec/`, `http://schema.org/`

[6] Example of a RDF wrapper request: `https://wrapper-url/ldw/oxford/search?query=Eugenio+Bonivento`



(a) Bipartite Graph (b) 1-1 Weighted Perfect Matching

Figure 6.6: **The 1-1 Weighted Perfect Matching Problem**. The algorithm to compute the 1-1 weighted perfect matching receives as input a weighted bipartite graph where weights represent the values of a similarity measure between the RDF molecules in the bipartite graph. The output of the algorithm is a maximal matching of the RDF molecules in the bipartite graph, where each RDF molecule is matched to exactly one RDF molecule; edges in the matching have a maximal value.

possible to attach additional semantic information to the KG, e.g., location information. Thus, the string “Italy” of a Twitter tweet can be annotated with resources from other knowledge graphs, such as DBpedia *Italy* resource[7]. Enrichment of on-demand KGs is achievable through facts mining based on the existing facts and using graph analysis algorithms. Additionally, two of the built-in advantages of the on-demand KGs built by *FuhSen* are: (1) provenance information, which allows to trace the origins of a certain fact to a certain source; and (2) the freshness of data, since web sources evolve over time the on-demand approach allows FuhSen to collect and integrate the latest data.

6.2.3 Integration of RDF Molecules

This module constructs a knowledge graph out of the enriched molecules. The input is a set of molecules, and the output is an integrated RDF graph. The module consists of three sub-modules, namely *Semantic Similarity* sub-module, *Perfect Matching* sub-module, and *Integration* sub-module. In this module we based on the results presented in Chapter 4. We describe below how we configure each sub-module of MINTE in details.

Computing Similarity of RDF Molecules

Similar molecules should be merged in order to create a fused, universal representation of a certain entity. In contrast with triple-based linking engines like Silk [158], we employ a molecule-based approach increasing the abstraction level and considering the semantics of molecules. That is, we do not work with independent triples, but rather with a set of triples belonging to a certain subject. The MINTE molecule-based approach (cf. Chapter 4) allows for natural clustering of a knowledge graph, reducing the complexity of the linking algorithm.

[7] http://www.dbpedia.org/resource/Italy

The 1-1 Weighted Perfect Matching

Given a weighted bipartite graph BG of RDF molecules, where weights correspond to values of semantic similarity between the RDF molecules in BG, a matching of BG corresponds to a set of edges that do not share an RDF molecule, and where each RDF molecule of BG is incident to exactly one edge of the matching. The problem of the 1-1 weighted perfect matching of BG corresponds to a matching where the sum of the values of the weights of the edges in the matching have a maximal value [159]. The *Hungarian algorithm* [160] computes the 1-to-1 weighted perfect matching. Figure 6.6(a) illustrates the input of the algorithm where BG comprises edges between RDF molecules, while Figure 6.6(b) represents the final state. RDF molecules with the maximal values of similarity are mapped in pairs in the solution of the 1-1 weighted perfect matching and will be considered as RDF molecules to be merged. To determine the minimal value of similarity that represents RDF molecules that may be considered similar, a threshold T in the range of [0.1] is considered. Edges with weights less than T are considered as 0.0 by the 1-1 weighted perfect matching algorithm.

Integration functions

When similar molecules are identified under the desired conditions, the last step of the pipeline is to integrate them into an RDF knowledge graph. The result knowledge graph contains all the unique facts of the analyzed set of molecules. The implementation of the integration function in *FuhSen* is the union, i.e., the logical disjunction, of the molecules identified as similar during the previous steps.

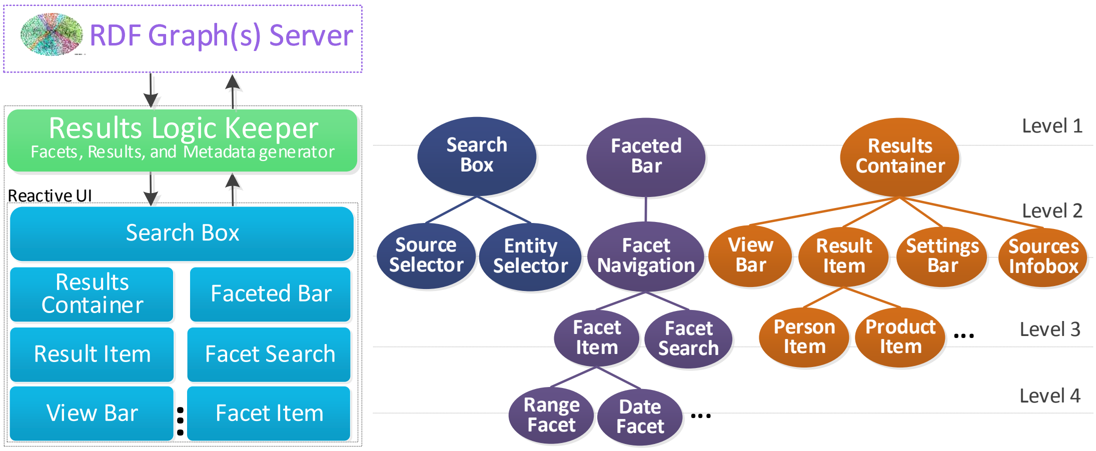
6.2.4 Exploration of RDF Molecules

Once a consolidated graph is built out of the web sources. The next step is to enable the exploration of the knowledge graph. The state-of-the-art user interfaces are mainly oriented to explore materialized knowledge graph and not on-demand created knowledge graphs. In this section, we show the design of a new approach to exploring knowledge graphs build on-demand from web sources named FaRBIE.

The FaRBIE Approach

To tackle the challenges of browsing on-demand built knowledge graphs, and to keep up with providing non-technical users with a more enjoyable and usable experience, we propose a *reactive user interface* design style. In contrast to imperative approaches (e.g., libraries such as jQuery), a Reactive UI updates itself by *reacting* to changes in the data and rendering the right components whenever such data changes occur. Reactive UIs are component-oriented, where each component may evolve independently, facilitating reusability in the interface. A reactive UI component may contain not only the *view* but also pieces of *logic* to react appropriately to the semantics of the data. Hence, we argue that this style of UI fits well for RDF-based applications. ReactJS[8] has become one of the most popular libraries to implement this style of user interface.

[8] https://facebook.github.io/react/



(a) *FaRBIE* architecture

(b) UI components

Figure 6.7: **UI Design.** (a) *FaRBIE* design contains the Results Logic Keeper and Reactive UI Components; (b) Reactive UI Components organized in levels from generic to specific. The UI components can be extended and specialized in providing a better UX according to the semantics of data.

User Interface Design

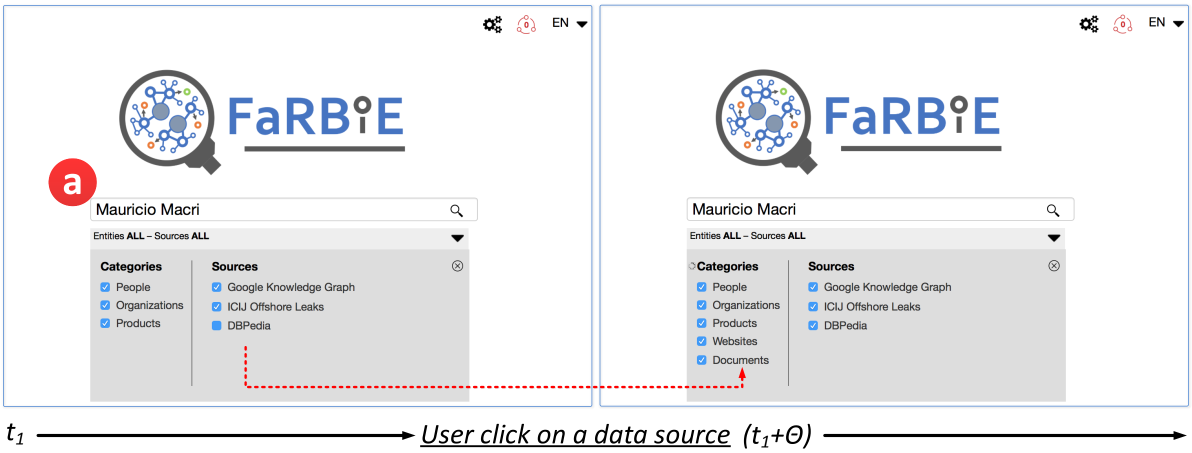
By making the user interface components reactive, we can provide developers with the possibility of making such components interact with the users in real time, while the queries for the multiple datasets are still running. In general, a common faceted browsing UI can be divided into three main UI sections: *i)* the search box or input section and source selection, where the user enters a keyword to start the search process and select data sources and entity types of interest. *ii)* the facets section contains all the UI elements to filter and narrow down the results; the facets are automatically generated based on the results collected from the RDF graphs. *iii)* the results section shows the entities found in the RDF graphs that match the keyword. Additionally, it provides users with feedback concerning the current state of a multiple dataset query (whether the search has succeeded, has failed, or is in progress). Figure 6.7 illustrates the *FarBIE* user interface design.

Logic Keeper

The Logic Keeper is not a user interface component but a component responsible for handling the communication with the RDF graph servers (e.g., using SPARQL HTTP requests). It uses the keyword search input and prepares the queries to the RDF graph servers. In this paper, we assume that the RDF graph server supports Keyword Search[9]. The Logic Keeper manages all the logic applied to the results coming from the RDF graph servers, including 1) facets generation, 2) entity results preparation, and 3) meta-data creation from the search process.

To *generate the facets*, we based our implementation on the work of Arenas et al. [35]. In brief, generic SPARQL queries are applied on the resulting entities to generate the facets. Additionally, the list of facets and the number of facet values are computed and maintained by the Logic Keeper. The second responsibility of the Logic Keeper is that of *preparing the entity results*, where a snippet is composed per result, and values such as dates are standardized for the reactive

[9] Common Triple Stores, such as Virtuoso or Fuseki, usually provide a keyword search functionality. https://jena.apache.org/documentation/query/text-query.html


Mauricio Macri

Mauricio Macri

Figure 6.8: (a) **Search Box** reactive component. A new data source selection produces a reaction in the categories list with the new entities available in the graph

UI components. Finally, *meta-data about the search process is created*, including provenance of results, server request success or error information, the number of results, the number of results by type and the type of entities. All this metadata is computed and managed by the Logic Keeper. To trigger the reaction of the UI components, the Logic Keeper communicates whenever there is a change in the search results data, under the following circumstances: 1) more results arrive from the RDF graphs, resulting in new facets or new result items, or 2) a user interaction in the UI components demands more data.

Reactive UI components

The following reactive components were designed to tackle the UI challenges mentioned in Section 6.1.2. For the sake of terminology, we will be naming entities as *categories* and attributes as *elements* in the following.

Search Box: The purpose of this UI component is to provide the user with an interface for the input of the query parameters, and thus the creation of the federated query. No particular reactivity needs to be incorporated in this component apart from the common auto-suggestion and auto-completion features well-know for this type of UI component.

Source & Entity Selector: Allows users to focus on specific web sources or entities. The *reactivity* designed on these components is triggered by a user interaction. When the user filters a data source, the categories list is updated with the searchable entities of that web source.

Faceted Bar: Through the use of a real-time-populated Faceted Bar, we attempt to address the *reactivity* and *visibility* issues in terms of views navigation and results (i.e. data) filtering for a set of RDF knowledge graph datasets. We attempt to achieve this by combining the *Accordion*[10] user interface element and the *List* menu patterns into one unified reactive component, which we call the Faceted Bar. The *Faceted Bar* provides users with a layered menu-styled navigational experience, in order to provide the ability of browsing through entities and thus filtering results as well, in a single component, which could potentially save time and improve the ergonomics. For the purposes of our design, we support the implementation of

[10] Accordion menu pattern in CSS3.
https://designmodo.com/css3-accordion-menu/

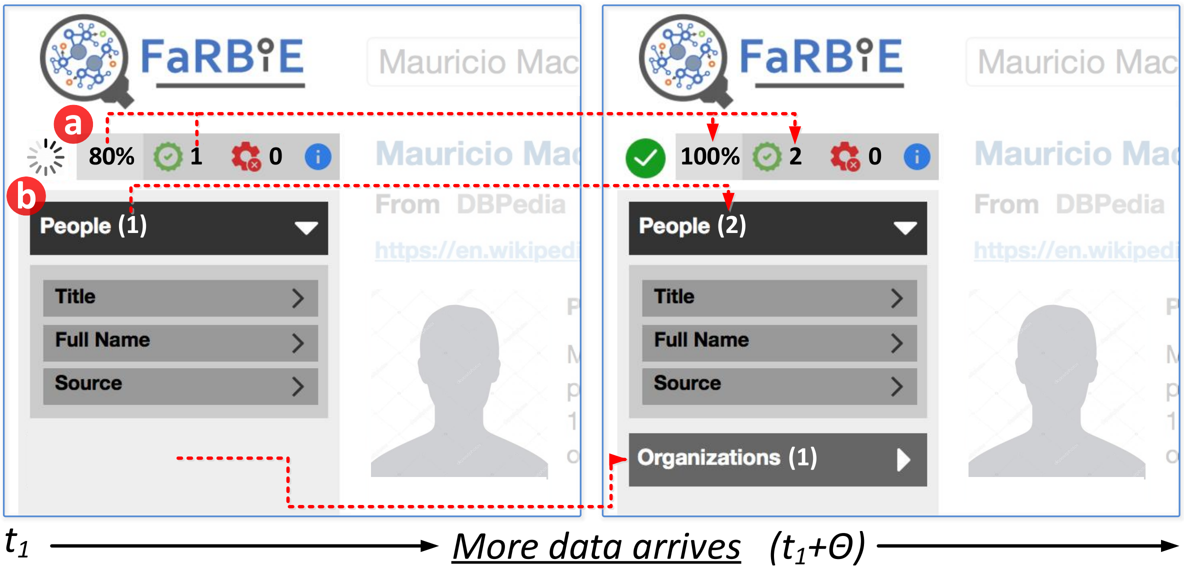


Figure 6.9: **(a) Source Box** component reacts when a data source query is retrieved successfully or with errors. **(b) Faceted Bar** component reacts to new facet items or categories.

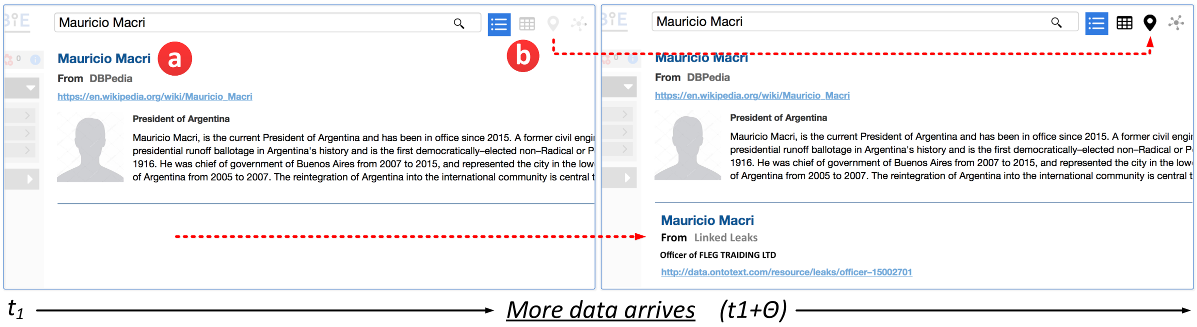
this component as a left-side-bar or as a top-bar, according to the developer's preference. The *Faceted Bar* also allows to minimize the impact of the *screen space trade-off*, for the benefit of saving additional screen space without sacrificing visibility in neither the navigation area (i.e., the categories and elements filtering) nor the actual screen space required for the data presentation (i.e., results presentation).

Facet Navigation Menu: The Facet Navigation Menu is a child UI component of the *Faceted Bar* UI component. It allows users to semantically navigate within the elements being obtained as results from the federated search query, through entity categorization. The idea is to provide menus and sub-menus which allow users to narrow down the list of results, in order to improve usability by means of offering better grouping and increased visibility, while providing a design backbone for future developers at the same time. Our approach supports default entity categorizations using the types *Persons, Organizations, Products*, and *Documents*. For instance, it maps all the elements obtained as a result of the query which belong to the *person* category, thus creating a "menu" container for such attributes. The Facet Navigation Menu UI component holds two child UI components: *Facet Item* and *Facet Search*.

Facet Items: The purpose of the Facet Item UI Component is to serve as the final display and selection place for each category. Thus, elements such as *gender* for a category of type *person*, for example, would be placed as checkbox UI components, which the user could use as selection to further narrow the results view of the query.

Facet Search: The purpose of the Facet Search UI Component is to serve as a means of a refined search among the obtained elements throughout each category. By using such component, we provide users also with the ability of refining the field of options available for navigation of the elements.

Sources Infobox: The purpose of this UI component is to provide a toolbox interface where users can have an overview of the status corresponding to the sources being queried. This component has three elements: 1) successfully retrieved sources, 2) failed sources, and 3) the *information button*, which pop ups a dialog with detailed information regarding the status of the



Mauricio Macri

Mauricio Macri

From DBPedia

https://en.wikipedia.org/wiki/Mauricio_Macri

President of Argentina

Mauricio Macri, is the current President of Argentina and has been in office since 2015. A former civil engi
presidential runoff ballotage in Argentina's history and is the first democratically–elected non–Radical or P
1916. He was chief of government of Buenos Aires from 2007 to 2015, and represented the city in the low
of Argentina from 2005 to 2007. The reintegration of Argentina into the international community is central t

Mauricio Macri

Mauricio Macri

From DBPedia

https://en.wikipedia.org/wiki/Mauricio_Macri

President of Argentina

Mauricio Macri, is the current President of Argentina and has been in office since 2015. A former civil engi
presidential runoff ballotage in Argentina's history and is the first democratically–elected non–Radical or P
1916. He was chief of government of Buenos Aires from 2007 to 2015, and represented the city in the low
of Argentina from 2005 to 2007. The reintegration of Argentina into the international community is central t

Mauricio Macri

From Linked Leaks

Officer of FLEG TRAIDING LTD

http://data.ontotext.com/resource/leaks/officer-15002701

Figure 6.10: **(a) Results Container** component reacts when more data arrive. Additionally, it selects the best view according to the semantics of the results. **(b) View Bar** component reacts after analyzing the results, e.g., the map view is enabled when geo-data is found in the search results.

sources, response times, as well as error codes and messages, which clearly indicate the nature of a failure in a queried source.

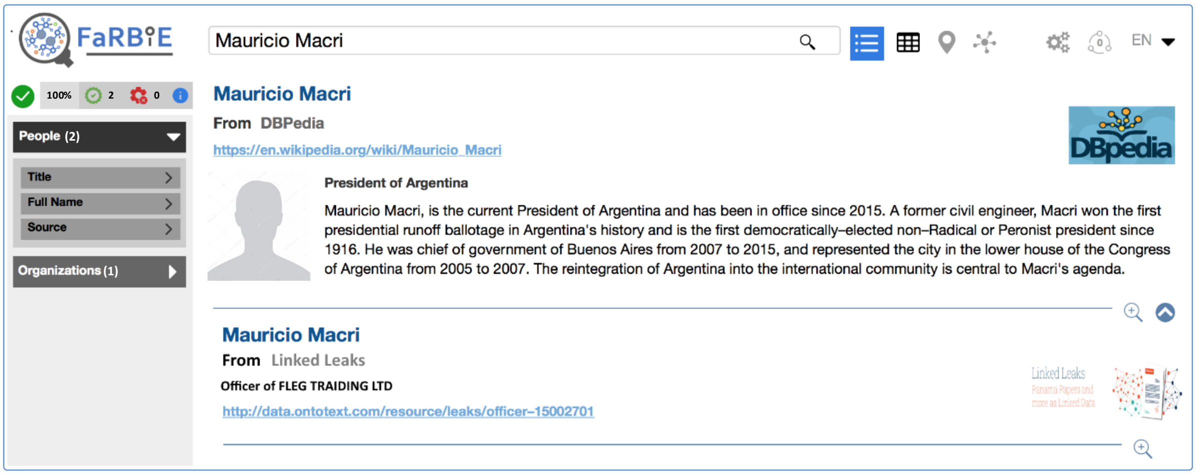
Results Container: The purpose of this UI component is to encapsulate the main UI components related to the results of the query. This includes the screen area allocated for displaying the list or map of results belonging to a query, in which each result of the query is then displayed in a child *ResultsItem* UI component. At the same time, the children *ResultsItem*, *Source Box*, *Views Bar* and *Settings Bar* can also be found under this UI component. The *reactivity* designed on this component is triggered whenever more data is coming from the RDF graphs, e.g., new results are appended to the list.

Results Item: The purpose of this UI component is to provide a unique container each of the results being obtained from the query will be mapped to, in order to be later displayed in the *Results Container* component. Thus, each *Results Item* UI component will be shown or hidden from the *Results Container*, depending on the navigational input obtained from the *Faceted Bar* component. The *reactivity* designed on this component is triggered by the semantics of the entities contained in the data, e.g., for a person, it may be more relevant to show demographic information, but for an organization, its location information might be more relevant. This is achieved by specialized views in the UI component structure hierarchy.

Views Bar: The purpose of this UI component is to provide the users with the possibility of switching the display of the results through different view modes. The *reactivity* designed on this component is triggered by more data coming from the RDF graphs. After analyzing the results, an appropriate view is automatically enabled. *FaRBIE* supports the following common *view modes*:

- *List mode*: Results are displayed in a list, it is the default view mode.
- *Table mode*: Results are shown in a table, each column is an entity attribute of interest.
- *Map mode*: Results are displayed on a map when geodata is provided.
- *Graph mode*: It uses a graph visualization to display the relationships between the results if a sufficient number of links is found.

Settings Bar: The purpose of this UI component is to provide the user with the possibility of accessing additional options, such as entering credentials to obtain log-in tokens, importing



Mauricio Macri

Mauricio Macri

From DBPedia

https://en.wikipedia.org/wiki/Mauricio_Macri

President of Argentina

Mauricio Macri, is the current President of Argentina and has been in office since 2015. A former civil engineer, Macri won the first presidential runoff ballotage in Argentina's history and is the first democratically–elected non–Radical or Peronist president since 1916. He was chief of government of Buenos Aires from 2007 to 2015, and represented the city in the lower house of the Congress of Argentina from 2005 to 2007. The reintegration of Argentina into the international community is central to Macri's agenda.

Mauricio Macri

From Linked Leaks

Officer of FLEG TRAIDING LTD

http://data.ontotext.com/resource/leaks/officer–15002701

Figure 6.11: **Proof of Concept.** FaRBIE allows to explore the on-demand built knowledge graph from DBpedia and Linked Leaks web sources.

a previously saved configuration file to use as input parameter for the federated search, or selecting the displayed language. Other options could be implemented and supported by future researchers and developers in order to adapt to their needs. No particular reactivity has been incorporated into this component.

Proof-of-concept

To validate our approach, we have developed a proof-of-concept, based on the introduced use case scenario in the criminal investigation domain. We have configured *FaRBIE* to explore two datasets, namely *DBPedia* and *Linked Leaks*. Figure 6.11 shows the results using *Mauricio Macri* as keyword. Two people and one organization were found matching the keyword, already providing insights to the possible activities and relation between *Mauricio Macri* and the Panama Papers scandal, with only a single search. In order to implement *FarBIE*, we evaluated different frameworks for web user interfaces as well as web development platforms resulting in the following selection:

- **ReactJS[11]:** A modern javascript library for building web user interfaces. It is a component-oriented library, and the features of virtual DOM it provides fit perfectly the requirements of modifying the user interface dynamically when new data is sent from the server to the user interface. In the work of Khalili et al. [118], it is used as the core technology to provide a reusable set of user interface elements to build Linked Data applications.
- **Web Socket:** A protocol providing full-duplex communication channels over a single TCP connection. It is an ideal protocol to realize the communication of the reactive user interface with the backend system. In our case, a web socket is opened between the LogicKeeper component and the federated search engine. The data is continuously pushed from the server to the client.
- **Play Framework[12]:** Is a high velocity web framework for Java and Scala. Many web

[11] https://facebook.github.io/react/

[12] https://www.playframework.com

	Experiment 1: People		Experiment 2: People			
	DBpedia D1	DBpedia D2	DBpedia	Wikidata	DBpedia	Wikidata
Molecules	500	500	500	500	1000	1000
Triples	17,951	17,894	29,263	16,307	54,590	29,138

Table 6.1: Benchmark Description. RDF datasets used in the evaluation.

frameworks, such as Grails, Tomcat, Spring, PHP, or Rails, use threaded servers. A threaded server assigns one thread per request and uses blocking I/O. The play framework is based on an event server (Netty). It assigns one thread/process per CPU core and uses non-blocking I/O. Threaded vs. event matters in a reactive user interface, as the engine spends most of the time waiting for query results.

The proof-of-concept interface and the intial source code is available as an open source project.[13] *FaRBIE* is empowered with flexible user interface components that react to new data coming from the server. The user is able to explore the portion of search results as soon as they are retrieved from the datasets in the RDF graph federation. Instant filtering is possible without waiting for the complete set of results.

6.3 Empirical Evaluations

6.3.1 Performance Evaluation

To answer research question 3 (cf. Section 1.3), we evaluate the effectiveness of FuhSen on building on-demand knowledge graphs using GADES—a semantic similarity metrics, compared to Jaccard—a non-semantic similarity metric. We assess the following research questions:

- **Q1:** Does a semantic similarity metric, i.e., GADES, synthesize RDF graphs on-demand more efficiently and effectively compared to Jaccard?
- **Q2:** What is the impact of threshold values on the completeness of the on-demand built knowledge graph?

The experimental configuration to evaluate these research questions is as follows:

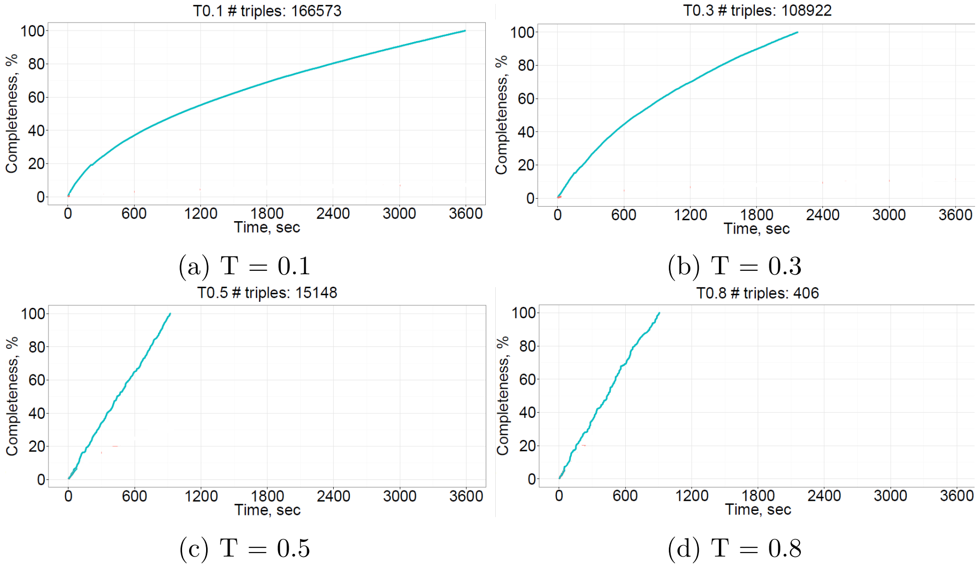
Experimental Setup

Benchmark: Experiment 1 is executed against a dataset of 500 molecules[14] of type Person extracted from the live version of DBpedia (February 2017). Based on the original molecules, we created two sets of molecules by randomly deleting or editing triples in the two sets sharing the same DBpedia vocabulary. Experiment 2 employs subsets of DBpedia and Wikidata of the Person class. Assessing FuhSen in the higher heterogeneity settings, we sampled datasets of 500 and 1000 molecules varying triples count from 16K up to 55K. Table 6.1 provides basic statistics on the experimental datasets.

Baseline: Gold standards include the original DBpedia Person descriptions (Experiment 1) and `owl:sameAs` links between DBpedia and Wikidata (Experiment 2). The Gold standard for

[13] https://github.com/LiDaKrA/FaRBIE

[14] https://github.com/RDF-Molecules/Test-DataSets/tree/master/DBpedia-People/20160819



(a) T = 0.1

(b) T = 0.3

(c) T = 0.5

(d) T = 0.8

Figure 6.12: **Experiment 1 (GADES) integrating molecules of DBpedia.** FuhSen produces complete results at all threholds.

evaluating FuhSen is comprised of the pre-computed amounts of pairs which similarity score exceeds a predefined threshold, the gold standards are computed offline.

Metrics: We report on execution time (ET in secs) as the elapsed time required by the FuhSen to produce all the answers. Furthermore, we measure *Completeness* over time, i.e., a fraction of results produced at a certain time stamp. The timeout is set to one hour (3,600 seconds), the operators results are checked every second. Ten thresholds in the range [0.1 : 1.0] and step 0.1 were applied in Experiment 1. In Experiment 2, five thresholds in the range [0.1 : 0.5] were evaluated because no pair of entities in the sampled RDF datasets has a GADES similarity score higher than 0.5.

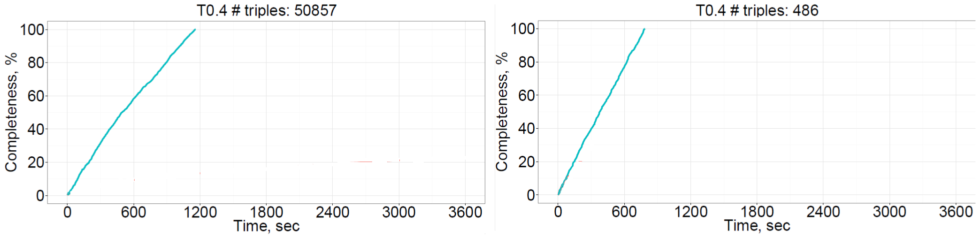
Implementation: For this experiment we implemented FuhSen using Scala and Play Framework[15]. The experiments were executed on a Ubuntu 16.04 (64 bits) Dell PowerEdge R805 server, AMD Opteron 2.4GHz CPU, 64 cores, 256GB RAM. We evaluated two similarity functions: GADES (cf. Section 5.2.2) and Jaccard (cf. Section 5.2.1). GADES relies on semantic descriptions encoded in ontologies to determine relatedness, while Jaccard requires the materialization of implicit knowledge and mappings. Evaluating schema heterogeneity of DBpedia and Wikidata in Experiment 2 the similarity function is fixed to GADES.

DBpedia to DBpedia People

Experiment 1 evaluates the performance and effectiveness of FuhSen. The testbed includes two split DBpedia dumps with semantically equivalent entities but non-matching resource URIs and randomly distributed properties; That is, both web sources are described in terms of one DBpedia ontology. GADES and Jaccard similarity functions are compared.

Figure 6.12 shows the results of the evaluation of FuhSen with GADES. FuhSen achieves completeness over time in all four cases with the threshold in the range 0.1-0.8. Figure 6.12(a) demonstrates that FuhSen is capable of producing 100% of results within the timeframe. In

[15] https://github.com/LiDaKrA/FuhSen-reactive



(a) T = 0.4, GADES

(b) T = 0.4, Jaccard

Figure 6.13: **Experiment 1 with fixed threshold**. GADES identifies two orders of magnitude more results than Jaccard while FuhSen still achieves full completeness.

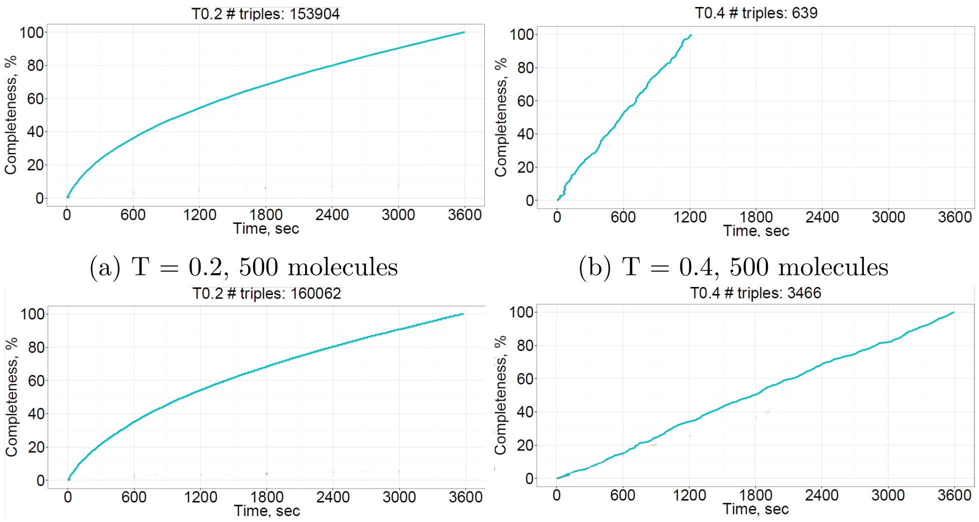
Figure 6.12(b), FuhSen achieves the full completeness even faster. In Figure 6.12(c) FuhSen finishes after 10 minutes. Figure 6.12(d) shows FuhSen taking a bit more time but achieves answer completeness. Figure 6.13 illustrates the difference in elapsed time and achieved completeness of FuhSen applying GADES and Jaccard similarity functions. Evidently, Jaccard outputs fewer tuples even on lower thresholds, e.g., 486 pairs at 0.4 threshold value, against 50,857 pairs by GADES. Analyzing the empirical results we are able to answer **Q1**, i.e., we demonstrate that plain set similarity metric as Jaccard that consider only an intersection of exactly same triples are ineffective in integrating heterogeneous RDF graphs. We also observe that FuhSen consistently exhibits reliable results. However, time efficiency depends on the input graphs and applied similarity functions. A further observation is that the semantic similarity function allows for matching RDF graphs more accurately.

DBpedia - Wikidata People

The distinctive feature of the experiment consists in completely different vocabularies used to semantically describe the same people. Therefore, traditional similarity metrics, e.g., Jaccard, are not applicable. Thus, we evaluate the performance of FuhSen employing GADES semantic similarity measure only. Results of FuhSen executed against 500 and 1000 molecules configurations are reported on 6.14. The observed behavior of FuhSen resembles the one in Experiment 1, i.e., FuhSen outputs complete results within a predefined time frame. Analyzing the observed empirical results, we are able to answer research questions **Q2**, i.e., a threshold value prunes the number of expected results and does not affect the completeness of FuhSen.

6.3.2 Usability Evaluation

This section presents the usability evaluations performed on FaRBIE, our on-demand built knowledge graph exploration approach. The goal is to ascertain that FaRBIE: a) allows to complete exploration tasks over on-demand built knowledge graph; b) is easy and pleasant to use compared to exploration interfaces of conventional knowledge graphs. To do so, first, we select two state-of-the-art user interfaces to compare our proposed approach, i.e., LD-R [118] and SemFacet [113, 114]. We used a formative evaluation technique and a usability evaluation questionnaire in a controlled environment. We selected 5 participants with experienced in software development. The participants were all male, aged 26-32.



(a) T = 0.2, 500 molecules

(b) T = 0.4, 500 molecules

(c) T=0.2, 1000 molecules

(d) T=0.4, 1000 molecules

Figure 6.14: **Experiment 2. FuhSen on-demand graph synthesization on different dataset sizes.** In larger setups, FuhSen still reaches full completeness.

Environment Set-up

A testing environment was set-up, in order to run all three systems in a stable and efficient matter. The evaluation was performed using a MacBook-Pro 2015 with 8GB of RAM, under the MacOS X High Sierra platform. In this regard, FaRBIE was tested through a web-server in the development environment. LD-R was evaluated using the live demo available under the project's homepage, while SemFacet and its dependencies were installed in a fresh, clean install of Ubuntu 16.04.03 LTS, installed under the aforementioned MacOS X High Sierra platform by means of the Oracle VM VirtualBox hypervisor software.

Formative Evaluation

A moderator introduced the experiments to the participants and controlled the task execution time. The evaluation instrument consisted of four simple tasks, each targeted at measuring task-usability per evaluated system:

1. Find information about a famous person the participant recognizes;
2. Find location information amongst the provided list of results;
3. Find an option in the system where to toggle the results display view, towards table- or map-based layouts;
4. Find information regarding errors that may have happened during the search process.

We define the following metrics: **Task Completion Rate**[16], a metric for measuring usability in terms of effectiveness, by means of the mathematical formula:

$$Effectiveness = \frac{SCT}{TNT} * 100\%$$

[16] https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/

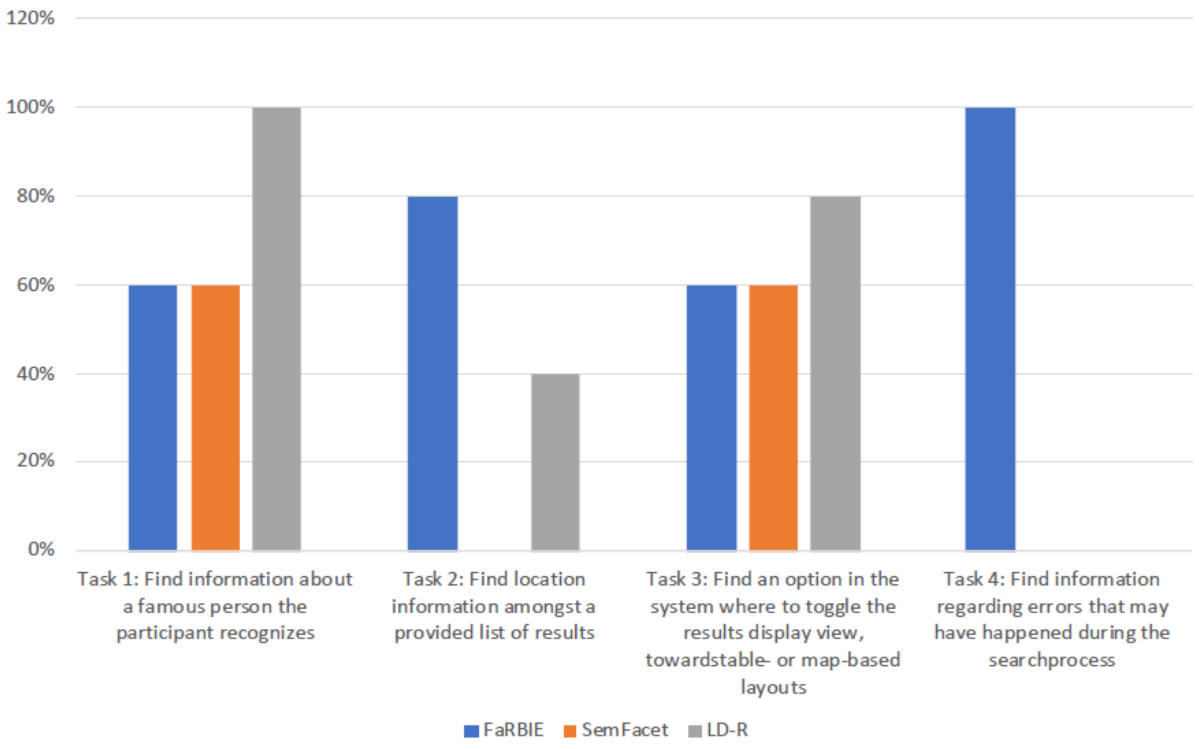


Figure 6.15: **FaRBIE formative evaluatio:** Overall Task Completion Rates.

with *SCT* as the number of *Succesfully Completed Tasks* per scenario, and *TNT* as the *Total Number of Tasks* per scenario. This metric allows for a rapid visualization of efficiency per task, with a reduced amount of effort—to notate the task completion rate per task per system per user, a single binary-notation system is used, where **1** indicates a successfully-completed task, with **0** indicating otherwise. At the end of the task, test, or evaluation, it just suffices to summarize the output results from this binary notation and calculate using the formula provided above. Next, while the test is underway, a task is marked as *not succesful* whenever the user surrenders from a task without having completed it.

Discussion: Figure 6.15 reports on the overall recorded **Task Completion Rates** for the usability evaluation. Users all four tasks at once per system, with a five minutes break between systems. The results exhibit FaRBIE operating within expected parameters, with many users scoring at least a completion rate per task of at least 60 percent, with it being the sole interface that managed to score a rate in Task 4, *Find information regarding errors that may have happened during the search process*, thus confirming an increase in visibility over other systems. The comparison systems did not perform as well as FaRBIE during the evaluation; **LD-R** came in second place performance-wise, faring better than FaRBIE in Task 1, *Find information about a famous person the participant recognizes* with a staggering 100 percent task-efficiency rate, as well as in Task 3 *Find an option in the system where to toggle the results display view, towards table or map-based layouts* with a reported task-efficiency rate of 80 percent. Meanwhile, SemFacet performed poorly, with a reported task-efficiency rate of less than 50 percent task completion rate in regards to its mean task-average.

Usability Evaluation

At the same time, and to gather supplementary usability information, the participants were asked questions with the help of a small questionnaire. The questionnaires were applied at the end of each system scenario testing cycle, in order to obtain valuable, non-quantitative additional feedback from the systems being tested. **After-Scenario Questionnaire (ASQ)**[17]: measured through qualitative methods that provide insights into additional feedback from users. This metric is evaluated through an on-site questionnaire. The questions included in this Likert-Scale questionnaire were as follows:

1. On a scale from 1-5, being 1 the most difficult and 5 the easiest, how easy-to-use did you find this website?

2. On a scale from 1-5, being 1 the most difficult and 5 the easiest, how did you find the navigation through this website?

3. Imagine this website is available to the public. On a scale from 1-5, being 1 the least probable, and 5 the most probable, how likely are you to using this website in the future?

4. In a scale of 1-5, being 1 the worst, and 5 the best, how well integrated did you find the functions in this website?

5. In a scale of 1-5, being 1 the least confident, and 5 the most confident, how confident did you feel using this website?

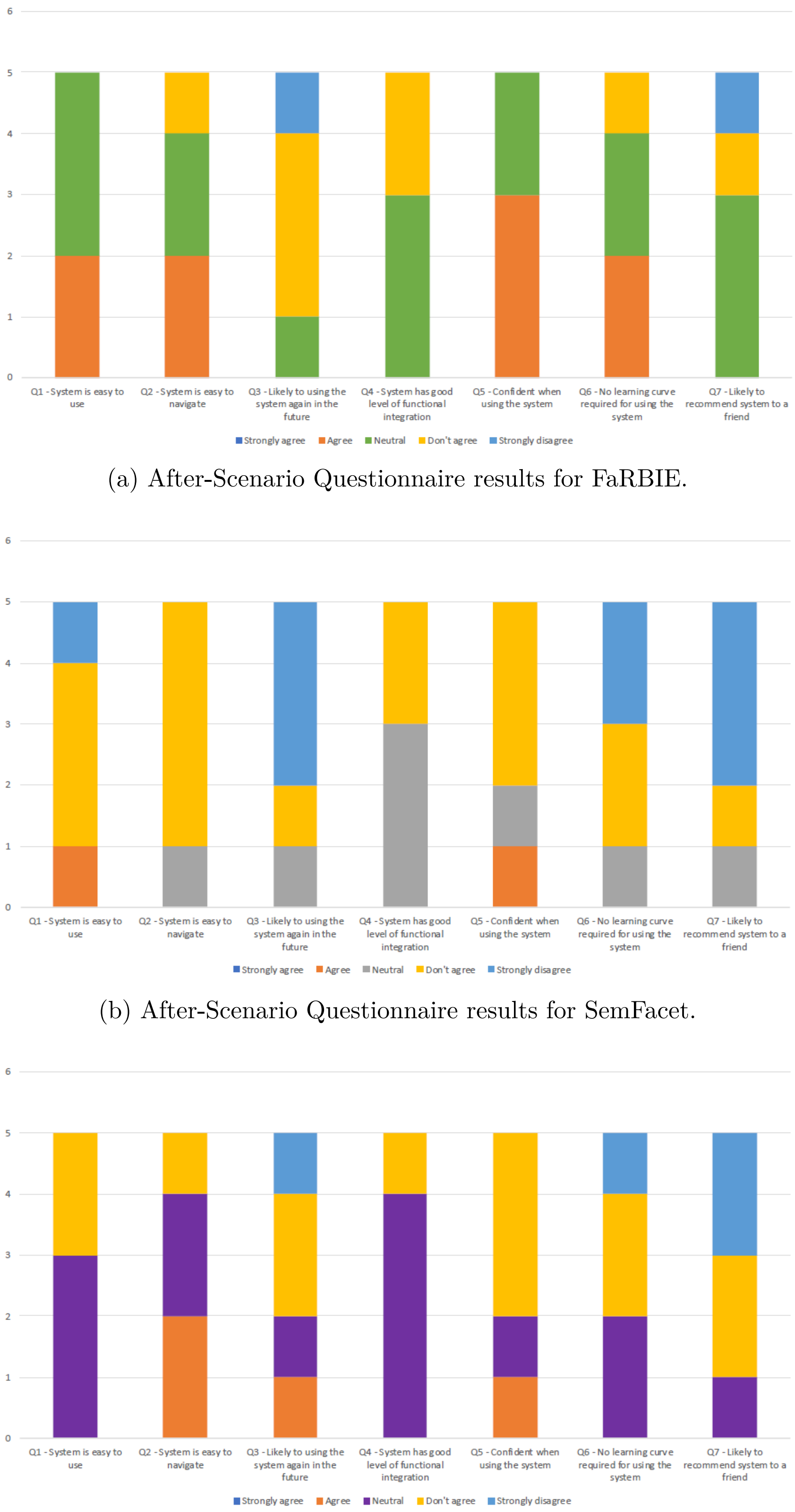
6. In a scale of 1-5, being 1 the slowest, and 5 the the fastest, how fast do you believe people would learn how to use this website?

7. In a scale of 1-5, being 1 the least likely, and 5 the most likely, how likely are you to recommend this website to a friend?

8. Please let us know any other remarks about this website that you feel important to share.

Discussion: Figure 6.16(a) reports on the feedback provided by users during the After-Scenario feedback collection. Users report feeling confident when using FaRBIE, while being neutral in all other categories except for re-using the software. However, since the system is still in prototype, it can be argued that these impressions should improve with a next, more stable release version of the interface. Nevertheless, SemFacet proved once more unpopular (Figure 6.16(b)), receiving only positive marks in relation to Q1 (likely to use the system again in the future) and Q7 (likely to recommend system to a friend), all the while it was the poorest performer in terms of task-completion rates. We observe similar results for LD-R (Figure 6.16(c)), with most of the participants being either neutral towards the system, or in strong disagreement of the statements. These statistics allow us to reach the conclusion that, while FaRBIE still has room for improvement, it ultimately shows promise as an *interface design pattern* for exploring on-demand built knowledge graphs, finally establishing a relevant design backbone that paves the way for on-demand build graph exploration.

[17] https://conversionxl.com/blog/8-ways-to-measure-ux-satisfaction/



(a) After-Scenario Questionnaire results for FaRBIE.

(b) After-Scenario Questionnaire results for SemFacet.

(c) After-Scenario Questionnaire results for LD-R.

Figure 6.16: **After-Scenario Feedback**. Analysis of the usability questionnaire: a) FaRBIE results; b) SemFacet results; and c) LD-R results.

6.4 Summary

In this chapter, we presented *FuhSen*, a federated hybrid search engine. *FuhSen* is able to create a knowledge graph on-demand by integrating data collected from a federation of heterogeneous web sources using an RDF molecule integration approach (MINTE). We have explained the creation of RDF molecules by using Linked Data wrappers; we have also presented how semantic similarity measures can be used to determine the relatedness of two entities in terms of the relatedness of their RDF molecules. Additionally, we presented *FaRBIE*, a reactive faceted browsing UI to explore on-demand built graphs. With the goal to provide a better user experience, *FaRBIE* follows a reactive user interface approach that handles the uncertainty in terms of connection, query response times, and size imposed by on-demand built graphs. *FaRBIE* is composed of several reactive UI components which react to changes of the semantics, variations in the size of the data, and the disparities in the response times coming from the different sources, allowing for a real-time user experience.

CHAPTER 7

Synthesizing Knowledge Graphs from Web Sources

In this chapter, we present the use of MINTE and FuhSen in three domain-specific applications. We name MINTE$^+$ to the implementation and combination of both approaches. In consequence, MINTE$^+$ is an integration framework that retrieves and integrates data from heterogeneous web sources into a knowledge graph. MINTE$^+$ implements novel semantic integration techniques that rely on the concept of RDF molecules to represent the meaning of this data; it also provides fusion policies that enable *synthesis* of RDF molecules. The content of this chapter is based on the publications [161, 162]. The results of this chapter provide an answer to the following research question:

> **RQ4**: How does semantic data integration impact the adaptability of knowledge retrieval systems?

We present the main results, showing a significant improvement of the task completion efficiency when the goal is to find specific information about an entity and discuss the lessons learned from each application. The remainder of the Chapter is structured as follows: First the MINTE$^+$ implementation is described in Section 7.1. Then, the application of MINTE$^+$ in Law Enforcement (Section 7.2), Job Marked Analysis (Section 7.3), and Manufacturing (Section 7.4) is described. Finally, Section 7.5 presents our summary and conclusions.

7.1 The Synthesis of RDF Molecules Using MINTE$^+$

Although several approaches and tools have been proposed to integrate heterogeneous data, a complete and configurable framework specialized for web sources is still not easy to set up. The power of MINTE$^+$ comes with the parameters to tune the integration process according to the use case requirements and challenges. MINTE$^+$ builds on the main outcomes of the semantic research community such as semantic similarity measures [7], ontology-based information integration, RDF molecules [121], and semantic annotations [163] to identify relatedness between entities and integrate them into a knowledge graph.

We are living in the era of digitization. Today as never before in the history of mankind, we are producing a vast amount of information about different entities in all domains. The

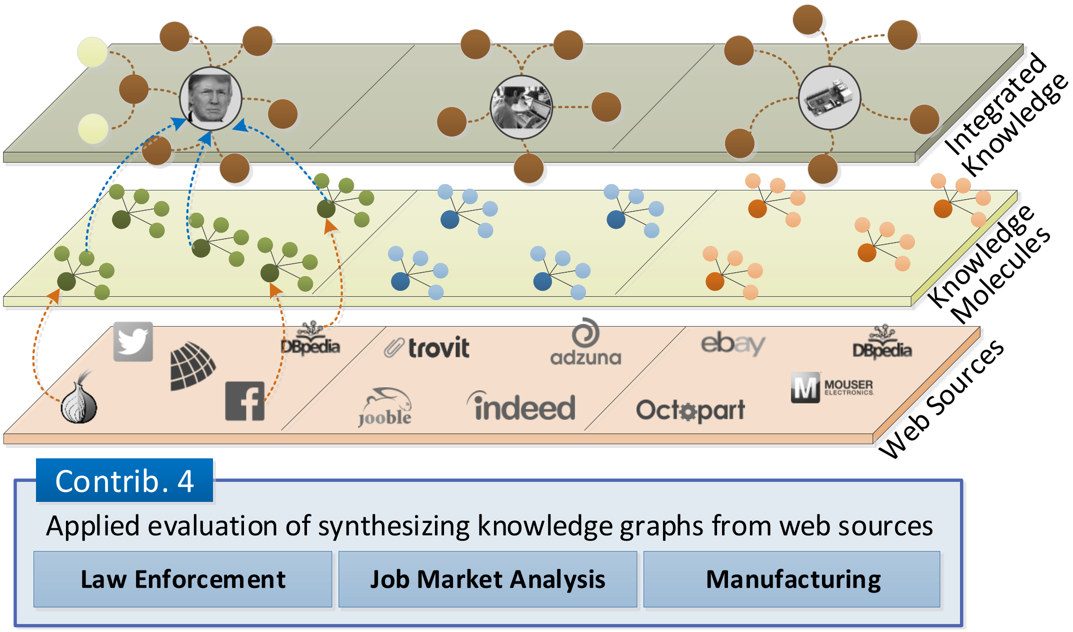


Figure 7.1: **Domain-specific applications**. (a) Law Enforcement agencies need to synthesize knowledge about suspects. (b) For a Job Market analysis, the job offers from different job portals need to be synthesized. (c) A manufacturing company needs synthesized knowledge about providers.

Web has become the ideal place to store and share this information. However, the information is spread across several web sources, with different accessibility mechanisms. The more the amount of information grows on the Web, the more important are efficient and cost-effective search, integration, and exploration of such information. Creating valuable knowledge out of this information is of interest not only to research institutions but to enterprises as well. Big companies such as Google or Microsoft spend a lot of resources in creating and maintaining so-called knowledge graphs. However, institutions such as law enforcement agencies, or SMEs cannot spend comparable resources to collect, integrate, and create value out of such data.

Institutions from different domains require the integration of data coming from heterogeneous Web sources. Typical use cases include Knowledge Search, Knowledge Building, and Knowledge Completion. We report on the implementation of MINTE$^+$ in three domain-specific applications: Law Enforcement, Job Market Analysis, and Manufacturing. The use of RDF molecules as data representation and a core element in the framework gives MINTE$^+$ enough flexibility to synthesize knowledge graphs in different domains. We first describe the challenges in each domain-specific application, then the implementation and configuration of the framework to solve the particular problems of each domain. We show how the parameters defined in the framework allow to tune the integration process with the best values according to each domain. Finally, we present the main results, and the lessons learned from each application.

Law enforcement agencies need to find information about suspects or illegal products on *web sites*, *social networks*, or private web sources in the Deep Web such as OCCRP[1]. For a job market analysis, job offers from different web portals need to be integrated to gain a complete view of the market. Finally, manufacturing companies are interested in information about their *providers* available in knowledge graphs such as DBpedia, which can be used to complete

[1] Organized Crime and Corruption Reporting Project, https://www.occrp.org/

the company's internal knowledge. Figure 7.1 illustrates the main problem and challenges of integrating pieces of knowledge from heterogeneous web sources. Although three different domain specific applications are presented, the core problem is shared: "synthesizing knowledge graphs from heterogeneous web sources", involving, for example, knowledge about *suspects*, or *job postings*, or *providers* (Layer 3 of Figure 7.1). This knowledge is spread across different web sources such as *social networks*, *job portals*, or Open Knowledge Graphs (Layer 1 of Figure 7.1). However, the integration of this information poses the following challenges:

- The lack of uniform representation of the pieces of knowledge.
- The need to identify semantically equivalent molecules.
- A flexible process for integrating these pieces of knowledge.

7.1.1 MINTE$^+$ Framework Implementation

Grounded on the semantic data integration techniques proposed in Chapter 4, the semantic similarity framework proposed in Chapter 5, and in the federated search engine proposed in Chapter 6. We implemented MINTE$^+$, an integration framework able to create, identify, and merge semantically equivalent RDF entities. Figure 7.2 depicts the main components of the MINTE$^+$ implementation. The pipeline receives a keyword-based query Q and a set of APIs of web sources (API_1, API_2, API_n) to run the query against. Additionally, the integration configuration parameters are provided as input. These parameters include: a semantic similarity measure Sim_f, a threshold γ, and an ontology O; they are used to determine when two RDF molecules are semantically equivalent. Furthermore, a set of fusion policies σ to integrate the RDF molecules is part of the configuration. MINTE$^+$ consists of three essential components: RDF molecule creation, identification, and integration. First, various RDF subgraphs coming from heterogeneous web sources are organized as RDF molecules, i.e., sets of triples that share the same subject. Second, the *identification component* discovers semantically equivalent RDF molecules, i.e., ones that refer to the same real-world entity; it performs two sub-steps, i.e., *partitioning* and *1-1 weighted perfect matching*. Third, having identified equivalent RDF molecules, MINTE$^+$'s semantic data integration techniques resemble the *chemical synthesis of molecules* [164], and the *integration component* integrates RDF molecules into complex RDF molecules in a knowledge graph.

7.1.2 Creating RDF Molecules

The *RDF molecule creation component* relies on search API methods, e.g., the API for searching people on Google+[2], and transforms an initial keyword-based query Q into a set of API requests understandable by the given web sources. MINTE$^+$ implements the mediator-wrapper approach; wrappers are responsible for physical data extraction, while a mediator orchestrates transformation of the obtained data into a knowledge graph. An ontology O provides formal descriptions for RDF molecules, using which the API responses are transformed into RDF molecules using SILK Transformation Tasks[3]. All the available sources are queried, i.e., no source selection technique is applied. Nevertheless, the execution is performed in an asynchronous fashion, so that the process requires as much time as the slowest web API. Once a request is

[2] https://developers.google.com/+/web/api/rest/latest/people/search

[3] http://silkframework.org/

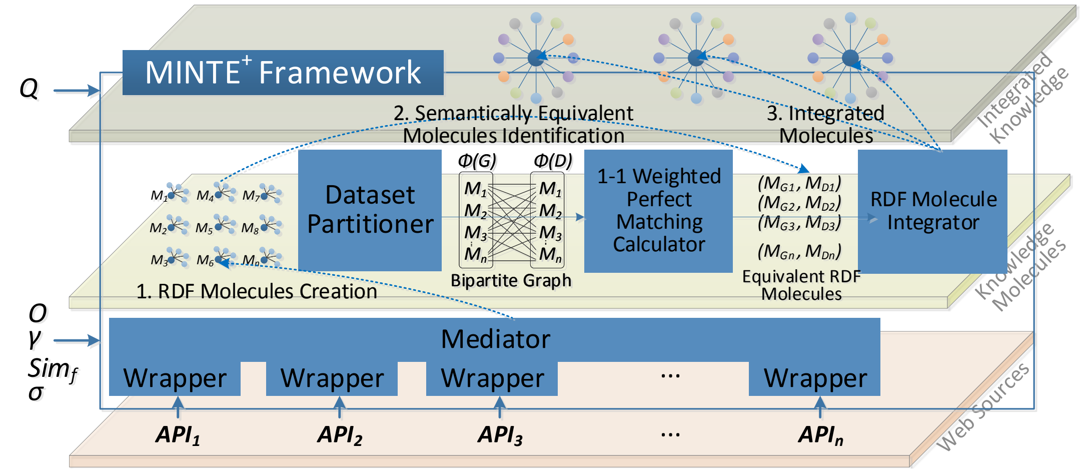
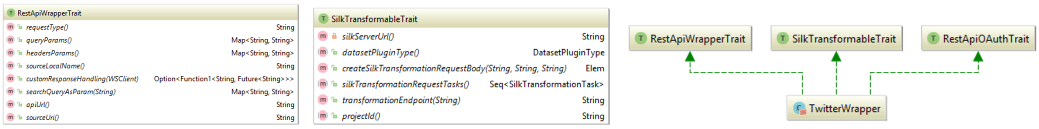


Figure 7.2: **The MINTE$^+$ Implementation.** MINTE$^+$ receives a set of web APIs, a keyword query Q, a similarity function Sim_f, a threshold γ, an ontology O, and a fusion policy σ. The output is a semantically integrated RDF graph.


```
RestApiWrapperTrait
requestType()                                                String
queryParams()                                   Map<String, String>
headersParams()                                 Map<String, String>
sourceLocalName()                                            String
customResponseHandling(WSClient) Option<Function1<String, Future<String>>>
searchQueryAsParam(String)                      Map<String, String>
apiUrl()                                                     String
sourceUri()                                                  String
```

(a) Web API Interface

```
SilkTransformableTrait
silkServerUrl()                                                      String
datasetPluginType()                                       DatasetPluginType
createSilkTransformationRequestBody(String, String, String)            Elem
silkTransformationRequestTasks()             Seq<SilkTransformationTask>
transformationEndpoint(String)                                       String
projectId()                                                          String
```

(b) SILK Interface

(c) Twitter Wrapper

Figure 7.3: MINTE$^+$ framework defines three basic interfaces for a wrapper: WebApiTrait, SilkTransformationTrait, and OAuthTrait.

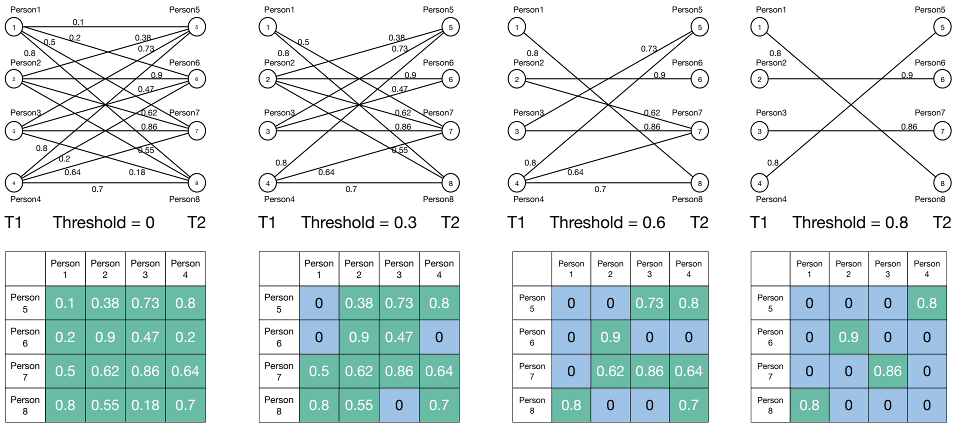
complete, wrappers transform the results into sets of RDF triples that share the same subject, i.e., RDF molecules. Then, the mediator aggregates RDF molecules into a knowledge graph, which is sent to the next component. These RDF molecule-based methods enable data transformation and aggregation tasks in a relatively simple way. Figure 7.3 depicts the interfaces implemented by a wrapper in order to be plugged into the pipeline.

7.1.3 Equivalent Molecules Identification

MINTE$^+$ employs a semantic similarity function Sim_f to determine whether two RDF molecules correspond to the same real-world entity, e.g., determining if two job posts are semantically equivalent. A similarity function has to leverage semantics encoded in the ontology O. For instance, GADES [7] implementation[4] supports this requirement. Additional knowledge about class hierarchy (`rdfs:subClassOf`), equivalence of resources (`owl:sameAs`), and properties (`owl:equivalentProperty`) enable uncovering semantic relations at the molecule level instead of just comparing plain literals. The identification process involves two stages: (a) dataset partitioning and (b) finding a perfect matching between partitions.

Dataset Partitioner. The partitioner component relies on a similarity measure Sim_f and an ontology O to determine relatedness between RDF molecules. Addressing flexibility, MINTE$^+$ allows for arbitrary, user-supplied similarity functions, e.g., simple string similarity and set

[4] https://github.com/RDF-Molecules/sim_service



	Person 1	Person 2	Person 3	Person 4
Person 5	0.1	0.38	0.73	0.8
Person 6	0.2	0.9	0.47	0.2
Person 7	0.5	0.62	0.86	0.64
Person 8	0.8	0.55	0.18	0.7

	Person 1	Person 2	Person 3	Person 4
Person 5	0	0.38	0.73	0.8
Person 6	0	0.9	0.47	0
Person 7	0.5	0.62	0.86	0.64
Person 8	0.8	0.55	0	0.7

	Person 1	Person 2	Person 3	Person 4
Person 5	0	0	0.73	0.8
Person 6	0	0.9	0	0
Person 7	0	0.62	0.86	0.64
Person 8	0.8	0	0	0.7

	Person 1	Person 2	Person 3	Person 4
Person 5	0	0	0	0.8
Person 6	0	0.9	0	0
Person 7	0	0	0.86	0
Person 8	0.8	0	0	0

Figure 7.4: **Bipartite Graph Pruning.** Various thresholds on a semantic similarity function and their impact on creating a bipartite graph between RDF molecules.

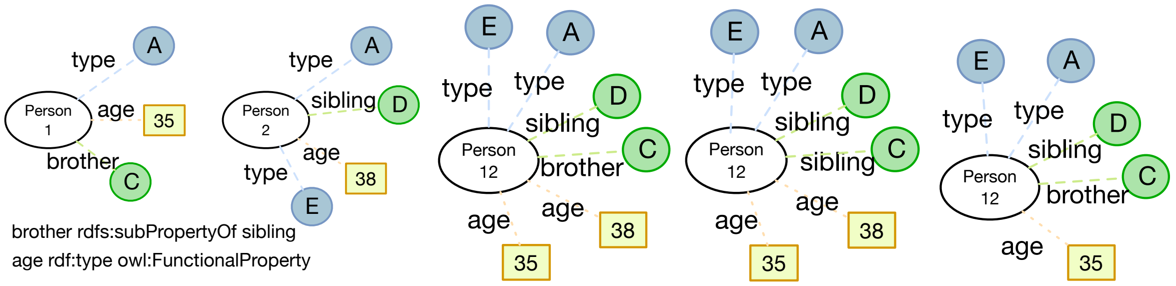
similarity. We, however, advocate for semantic similarity measures as they achieve better results (as we show in Chapter 4) by considering semantics encoded in RDF graphs. After computing similarity scores, the partitioner component constructs a bipartite graph between the sets of RDF molecules; it is used to match the RDF molecules.

A threshold γ bounds the values of similarity when two RDF molecules cannot be considered similar. It is used to prune edges from the bipartite graph whose weights are lower than the threshold. Figure 4.6 illustrates how different threshold values affect the number of edges in a bipartite graph. Low threshold values, e.g., 0, result in graphs with almost all the edges. Contrarily, when setting a high threshold, e.g., 0.8, graphs are significantly pruned.

1-1 Weighted Perfect Matching. Having prepared a bipartite graph in the previous step, the *1-1 weighted perfect matching component* identifies the equivalent RDF molecules by matching them with the highest pairwise similarity score; a Hungarian algorithm is used to compute the matching. Figure 7.4 (γ=0.8) illustrates the result of computing a 1-1 weighted perfect matching on the given bipartite graph. MINTE$^+$ demonstrates better accuracy when semantic similarity measures like GADES are applied when building a bipartite graph.

7.1.4 RDF Molecule Integration

The third component of MINTE$^+$, namely the RDF molecule *integration component*, leverages the identified equivalent RDF molecules in creating a unified knowledge graph. In order to retain knowledge completeness, consistency, and address duplication, MINTE$^+$ resorts to a set of *fusion policies* σ implemented by rules that operate on the RDF triple level. These rules are triggered by a certain combination of predicates, objects, and axioms in the ontology O. Fusion policies resemble flexible filters tailored for specific tasks, e.g., keep all literals with different language tags or retain an authoritative one, replace one predicate with another, or simply merge all predicate-value pairs of given molecules. Ontology axioms are particularly useful when resolving conflicts and inequalities on different semantic levels. Types of fusion policies include the following: Policies that process RDF resources such as dealing with URI naming conventions, are denoted as a subset $\sigma_r \in \sigma$. Policies that focus on properties are denoted as $\sigma_p \in \sigma$. Interacting with the ontology O, σ_p tackles property axioms, e.g., `rdfs:subPropertyOf`, `owl:equivalentProperty`, and `owl:FunctionalProperty`. Property-level fusion policies tackle



(a) Equivalent RDF Molecules (b) Union (c) Subproperty (d) Functional

Figure 7.5: **Merging Semantically Equivalent RDF Molecules**. Applications of a fusion policy σ: (a) semantically equivalent molecules R_1 and R_2 with two ontology axioms; (b) simple union of all triples in R_1 and R_2 without tackling semantics; (c) p_3 is replaced as a subproperty of p_4; (d) p_2 is a functional property and R_1 belongs to the authoritative graph; therefore, literal C is discarded.

sophisticated OWL restrictions on properties. That is, if a certain property can have only two values of some fixed type, σ_p has to guide the fusion process to ensure semantic consistency. Lastly, the policies dedicated to objects (both entities and literals) comprise a subset $\sigma_v \in \sigma$. On the literal level, the σ_v policies implement string processing techniques, such as recognition of language tags, e.g., *@en*, *@de*, to decide whether those literals are different or contain synta. For object properties, the σ_v policies deal with semantics of the property values, e.g., objects of different properties are linked by `owl:sameAs`. In this application of MINTE$^+$, the following policies are utilized [121]:

Union policy. The union policy creates a set of $(prop, val)$ pairs where duplicate pairs, i.e., pairs that are syntactically the same, are discarded retaining only one pair. In Figure 7.5(a) the pair $(type, A)$ appears in both molecules. In Figure 7.5(b), only one pair is retained. The rest of the pairs are added directly.

Subproperty policy. The policy tracks if a property of an RDF molecule is annotated as `rdfs:subPropertyOf`. As a result of applying this policy, the more general property is kept. The default σ_v object policy is to keep the property value of p_1 unless a custom policy is specified. In Figure 7.5(c), a property *brother* is generalized to *sibling* preserving the value C according to the subproperty ontology axiom in Figure 7.5(a).

Authoritative graph policy. The policy selects one RDF graph as a major source when merging various configurations of $(prop, val)$ pairs:

- The **functional property policy** keeps track of the funcional properties annotated as `owl:FunctionalProperty`, i.e., such properties may have only one value. The authoritative graph policy then retains the value from the primary graph: $\{r_1, p_1, B\}, \{r_2, p_1, C\} + O + functional(p_1) \models \{\sigma_r(r_1, r_2), p_1, \sigma_v(B, C)\}$. Annotated as a functional property in Figure 4.8(a), *age* has the value 35 in Figure 4.8(d), as the first graph has been marked as authoritative beforehand. The value 38 is therefore discarded.

- The **equivalent property policy** is triggered when two properties of two molecules are equivalent, i.e., they are annotated as `owl:equivalentProperty`: $\{r_1, p_1, A\}, \{r_2, p_2, B\} + O + equivalent(p_1, p_2) \models \{\sigma_r(r_1, r_2), \sigma_p(p_1, p_2), \sigma_v(A, B)\}$. The authoritative policy selects a property from the authoritative graph, e.g., either p_1 or p_2. By default, the property value is taken from the chosen property. Custom σ_v policies may override these criteria.

- The **equivalent class or entity policy** contributes to the integration process when entities are annotated as `owl:equivalentClass` or `owl:sameAs`, i.e., two classes or individuals represent the same real-world entity, respectively: $\{r_1, p_1, A\}, \{r_2, p_2, B\} + O + \mathit{equivalent}(A, B) \models \{\ \sigma_r(r_1, r_2),\ \sigma_p(p_1, p_2),\ \sigma_v(A, B)\}$. Similarly to the equivalent property case, the value with its corresponding property is chosen from the primary graph. Again, custom σ_p policies may handle the merging of properties.

7.2 Law Enforcement Application

7.2.1 Motivation and Challenges

Law enforcement agencies and other organizations with security responsibilities are struggling today to capture, manage and evaluate the amounts of data stored in countless heterogeneous web sources. As Figure 7.1a shows, possible sources include the document-based Web (so-called "visible net"), usually indexed by search engines such as Google or Bing. The Social Web (e.g., Facebook or Twitter), the Deep Web and the Dark Web (so-called "invisible net"). Deep web sources, such as e-commerce platforms (e.g., Amazon or eBay), cannot be accessed directly, but only via web interfaces e.g., REST APIs. The same holds for dark web sources, which are usually among the most relevant web sources for investigating online crime. Finally, open data catalogs in the Data Web, i.e., machine-understandable data from open sources such as Wikipedia, serve as sources of information for investigations. Law enforcement agencies spend a lot of time on searching, collecting, aggregating, and analyzing data from heterogeneous web sources. The main reason for such inefficient knowledge generation is that the agencies need different methods and tools to access this diversified information. If the investigators are not experts in a particular tool or technique, such as querying the Web of Data using SPARQL, they may not find the information they need. Thus, there is a lack of a holistic overview of the entities of interest. Without knowledge of programming, APIs, query languages, data analysis, etc., an investigator is not able to use and link all available data sources. Finally, most current search technology is based on simple keywords but neglects semantics and context. The latter is particularly important if you are looking for people with common names such as "Müller" or "Schmidt". Here, a context of related objects such as other people, places or organizations is needed to make a proper distinction. The main challenges of this application are the following:

C1. Heterogeneity of accessibility: Different access mechanisms need to be used to collect data from the web sources. Social networks require user-token authentication, deep web sources use access keys, and dark web sources require the use of the special software Tor Proxy[5].

C2. Provenance Management: Law enforcement institutions need to know the origin of the data, for a post-search veracity evaluation.

C3. Information Completeness: Although the process should be as automatic as possible, no data should be lost, e.g., all aliases or names of a person should be kept.

C4. Privacy by design: The system must be fully compliant with data protection laws, e.g., the strict ones that hold in the EU and especially in Germany. Citizens privacy is mainly protected by a fundamental design decision: No comprehensive data warehouse is built-up, but information is access on-demand from the Web sources.

[5] https://www.torproject.org/

Parameter	Value	Description
Query	Free Text	usually people, organizations, or products name or description.
Ontology	LiDaKrA	the ontology describing the main concepts in the crime investigation domain
Web APIs	11	Facebook, Google+, VK, Twitter, Xing, ICIJ Offshore Leaks, DBpedia, eBay, darknet sites, crawled darknet markets, OCCRP reports
Simf	GADES [7]	A semantic similarity measure for entities in knowledge graphs
Threshold	0.9	Only highly similar molecules are synthesized.
Fusion Policy	Union	No information is lost, e.g., all alias names of a person are kept in the final molecule.

Table 7.1: **MINTE$^+$ Configuration**. The Law Enforcement Application

The LiDaKrA[6] project has as main goal the implementation of a Crime Analysis Platform to solve the challenges presented above. The platform concept should be offered as a platform-as-a-service intended to support police departments, in the following use cases:

U1. Politically Exposed Persons: searching for politicians' activity in social networks, and possible relations with corruption cases and leaked documents detailing financial and client information of offshore entities. Relevant sources are Google+, Twitter, Facebook, DBpedia, OCRRP, Linked Leaks[7], etc.

U2. Fanaticism and terrorism: searching for advertising, accounts and posts on social networks. Relevant sources are Twitter, Google+, OCRRP, etc.

U3. Illegal medication: searching for web sites, posts, or video ads, with offers or links to darknet markets. Relevant sources are darknet markets, Tweets, Facebook posts, YouTube videos, ads, etc.

7.2.2 MINTE$^+$ Configuration

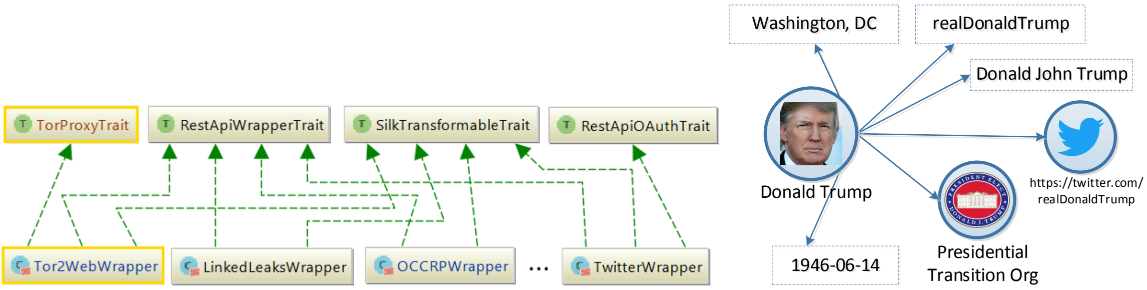
In order to solve these challanges we configured MINTE$^+$ in the following way: To address the challenges of this application and support the use cases, we configured MINTE$^+$ with the parameters shown in Table 7.1. As keyword Q, the users mainly provide people, organization, or product names, e.g., Donald Trump, Dokka Umarov, ISIS, or Fentanyl. Figure 7.7(a) shows the main RDF molecules described with the LiDaKrA domain-specific ontology O developed for this application. To address C1, thirteen wrappers were developed by implementing the interfaces described in Figure 7.6(a). These interfaces were sufficient for the social network and deep web sources defined in the application. However, an extension to access dark web sources was needed. A new interface was defined to enable a wrapper to connect to the Darknet using a Tor Proxy. As the similarity function, we used GADES [7] with a threshold of 0.9. This high value guarantees that only very similar molecules are integrated.

To address C2, each RDF molecule is annotated with its provenance at creation time using PROV-O[8], Figure 7.6(b) shows an RDF molecule example. The fusion policy *Union* was selected to address C3; this guarantees no information is lost during the integration process, e.g., whenever a person has two aliases, both are kept in the final molecule. By design, MINTE$^+$ does not persist any result in a triple store. All molecules are integrated on demand and displayed to the user. The on demand approach addresses challenge C4.

[6] https://www.bdk.de/der-bdk/aktuelles/artikel/bdk-beteiligt-sich-im-forschungsprogramm-lidakra

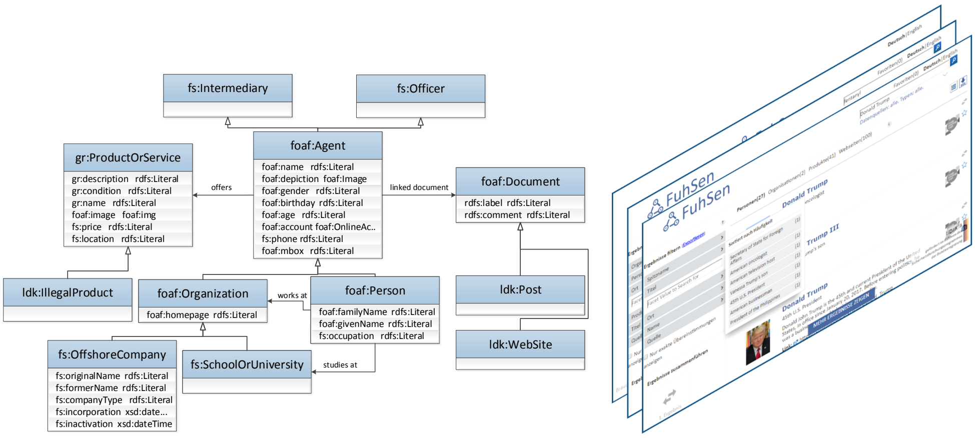
[7] http://data.ontotext.com/

[8] https://www.w3.org/TR/prov-o/



(a) Wrapper Extension for Tor (b) Functional

Figure 7.6: **MINTE$^+$ in the Law Enforcement Application**. (a) A new wrapper interface is implemented for querying the Dark Web. (b) An RDF molecule synthesized by the application; it synthesizes information about Donald Trump.



(a) The LiDaKrA ontology (b) User interface

Figure 7.7: **MINTE$^+$ in LiDaKrA**. (a) LiDaKrA UML ontology profile view (cf. [165]) of the main RDF molecule types. (b) The faceted browsing user interface that allows the exploration of the synthesized RDF molecules.

To close the application cycle, a faceted browsing user interface exposes the integrated RDF graph to users. Figure 7.7(b) shows the UI; users *pose* keyword queries and *explore* results using a multi-faceted browsing user interface. We chose facets as a user-friendly mechanism for exploring and filtering a large number of search results [35]. In Chapter **??**, we presented a demo of the user interface, comprising the following elements: a text box for the search query, a result list, entity summaries, and a faceted navigation component. Technically, MINTE$^+$ provides a REST API to execute its pipeline on demand. JSON-LD is the messaging format between the UI and MINTE$^+$ to avoid unnecessary data transformations for the UI components.

7.2.3 Results and Lessons Learned

Currently, the application is installed in *four law enforcement agencies* in Germany for evaluation.[9] The user feedback is largely positive. The use of semantics in the integration process and as input for the faceted navigation gives the necessary context to facilitate the exploration

[9] For confidentiality we cannot state their names, nor gather usage data automatically.

and disambiguation of results, e.g., suspects with similar names. One main user concern about the application relates to the completeness of results, e.g., a person is not found by MINTE$^+$ but it is found via an interactive Facebook search. Since MINTE$^+$ is limited to the results returned by the *API*, completeness of results cannot be guaranteed.

Thanks to MINTE$^+$, law enforcement agencies can integrate new web sources into the system with low effort (1–2 person days). This dynamicity is important in this domain due to some web sources going online or offline frequently. The users gave further important on the possibility to integrate internal data sources of the law enforcement agencies into the framework, which is possible thanks to the design of MINTE$^+$. The keyword search approach allows MINTE$^+$ to cope with all use cases defined for the system (e.g., U1, U2, and U3). In this application, we validate that the MINTE$^+$ framework works in an on-demand fashion. The main result of this application has become a product offered by Fraunhofer IAIS, which shows the maturity of MINTE$^+$'s approach.[10]

7.3 A Job Market Application

7.3.1 Motivation and Challenges

Declared by Harvard Business Review as the "sexiest job of the 21st-century"[11], data scientists and their skills have become a key asset to many organizations. The big challenge for data scientists is making sense of information that comes in varieties and volumes never encountered before. A data scientist typically has a number of core areas of expertise, from the ability to operate high-performance computing clusters and cloud-based infrastructures, to apply sophisticated big data analysis techniques and produce powerful visualizations. Therefore, it is in the interest of all companies to understand the *job market* and the *skills* demand on this domain. The main goal of the European Data Science Academy (EDSA), which was established by an EU-funded research project and will continue to exist as an "Online Institute"[12], is to deliver learning tools that are crucially needed to close this problematic skills gap. One of these tools is a dashboard intended for the general public, such as students, training organizations, or talent acquisition institutions. Through this dashboard, users can monitor trends in the job market and fast evolving skill sets for data scientists. A key component of the dashboard is the demand analysis responsible for searching, collecting and integrating job postings from different job portals. The job posts need to be annotated with the skills defined in the SARO ontology [163] and enriched with geo-location information; it presents the following challenges:

C1. Complementary Information: A complete view of the European data science job market is needed by gathering job postings from all member states.

C2. Information Enrichment: The job posting description should be annotated with the required skills described in the text.

C3. Batch Processing: To get an updated status of the job market, job postings should be extracted at least every two weeks.

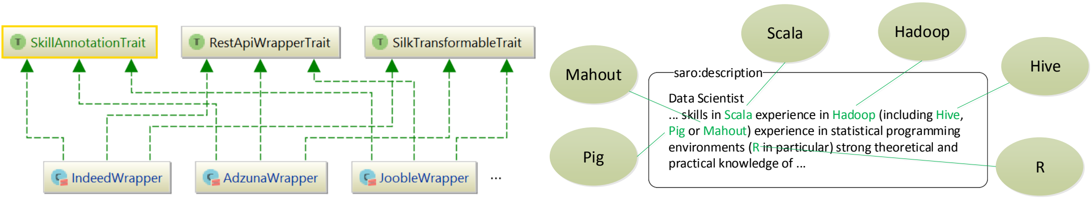
[10] https://www.iais.fraunhofer.de/de/geschaeftsfelder/enterprise-information-integration/uebersicht/dezentrale-suche.html

[11] https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century

[12] http://edsa-project.eu/

Parameter	Value	Description
Query	Job Title + Country	list of 150 job titles, e.g., Machine Learning, and 28 EU Countries, e.g., IT (Italy)
Ontology	SARO [163]	The ontology describes data scientist job postings and skills.
Web APIs	5	Adzuna, Trovit, Indeed, Jooble, and XING
Simf	SILK [78]	Job title, description and hiring organization are used in the linking rules.
Threshold	0.7	best score to integrate the same job posting from different job portals
Fusion Policy	Authoritative	Adzuna was defined as the main source.

Table 7.2: **MINTE$^+$ Configuration**. The Job Market Analysis Application


SkillAnnotationTrait
RestApiWrapperTrait
SilkTransformableTrait
IndeedWrapper
AdzunaWrapper
JoobleWrapper
...
Scala
Hadoop
Mahout
Hive
Pig
R
saro:description
Data Scientist
... skills in Scala experience in Hadoop (including Hive, Pig or Mahout) experience in statistical programming environments (R in particular) strong theoretical and practical knowledge of ...

(a) Skill annotation wrapper extension

(b) RDF Molecule

Figure 7.8: **MINTE$^+$ in the Job Market Application**. (a) A new wrapper interface is implemented for annotating a job description with the corresponding skills defined in the SARO ontology. (b) An RDF molecule synthesized by the application; it synthesizes an annotated job description.

The EDSA dashboard uses the results of the MINTE$^+$ integration framework; it can address the following use cases:

U1. Searching for a job offer: Search for relevant data scientist jobs by EU country or based on specific skills (e.g., Python or Scala).

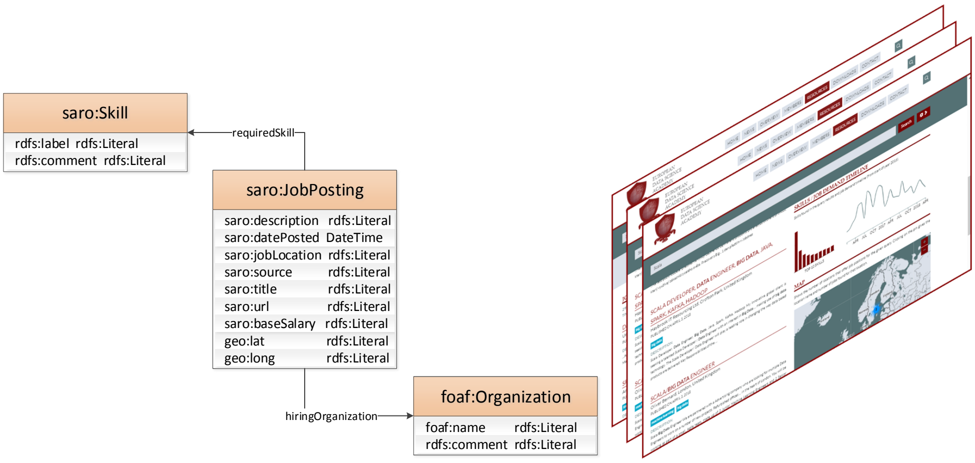
U2. Missing Skills Identification: it should be possible to identify what skills a person is missing on their learning path to becoming a data scientist.

U3. Analysis of Job Market By Country: analyze which EU country has more job offers, what is the average salary per country, etc.

U4. Top 5 Required Skills: identify the current top 5 relevant skills for a data scientist.

7.3.2 MINTE$^+$ Configuration

To address the stated challenges and to support the use cases, we configured MINTE$^+$ with the parameters shown in Table 7.2. A query Q is constructed from a list of 150 job titles and 28 countries. The combination of both is used as a keyword, e.g., "Machine Learning IT", yielding a total of 4,200 results. Figure 7.9(a) depicts the RDF molecule described with the SARO ontology O [163]. To address C1, five wrappers (Adzuna, Trovit, Indeed, Jooble, and XING) were developed by implementing the interfaces described in Figure 7.8(b). The data sources were selected covering as many countries as possible, e.g., Adzuna provides insights on the DE, FR, UK, IT markets. Indeed complements with data from NL, PL, ES. To address C2, a new interface *SkillAnnotationTrait* was defined. Figure 7.8(a) shows how the wrappers implement



(a) SARO ontology

(b) User Interface

Figure 7.9: **MINTE$^+$ in EDSA**. (a) The SARO ontology defines the RDF molecules for job market analysis. (b) Screenshot of the EDSA dashboard.

this new interface in addition to the standard ones defined in the framework. Technically, we employ GATE Embedded[13] to do the annotation using the SARO ontology.

As a similarity function, we resort to SILK [78] with a threshold of 0.7. The threshold was assigned after an empirical evaluation of the linkage rules in SILK. The RDF molecules created from job posts are similar in terms of properties. The Authoritative fusion policy was configured in this scenario, as only one property is required for fusion. Adzuna was defined as a main source. To periodically extract and integrate the job postings, a script was developed. The script reads the file containing the list of job titles and countries, calls MINTE$^+$ through its API, and saves the results in a triple store. Thus, batch processing (challenge C3) is addressed. Then, the EDSA dashboard shows the integrated information about the EU job market.

7.3.3 Results and Lessons Learned

The EDSA dashboard[14] is running and open to the general public. Thanks to the flexibility of the wrappers, the skills annotation behavior was easy to implement. The integrated job posting knowledge graph serves as the information source to address the defined use cases (U1, U3) by using the dashboard. Using a semantic representation of job postings, it was feasible to link the job market analysis with the supply analysis (i.e., the analysis of learning material) and the learning path identified in use cases U2 and U4. The main conclusion on this application is that MINTE$^+$ is able to support an intense integration process (batch mode). Overall, it takes one day to execute all the query combinations and update the status of the job market.

[13] https://gate.ac.uk/family/embedded.html

[14] http://edsa-project.eu/resources/dashboard/

Parameter	Value	Description
Query	Provider metadata	includes company name, address, web site.
Ontology	Schema.org	An extension of organization concept is used to describe the providers.
Web APIs	4	DBpedia, Google Knowledge Graph, plus further confidential sources
Simf	SILK	Wikipedia page is used in the linking rule.
Threshold	1.0	Providers with same Wikipedia page are integrated.
Fusion Policy	Authoritative	DBpedia is defined as the main source.

Table 7.3: **MINTE$^+$ Configuration**. The Manufacturing Application

7.4 Smart Manufacturing Application

7.4.1 Motivation and Challenges

The application is motivated by a global manufacturing company[15], which needs to complement their internal knowledge about parts providers with external web sources. The final usage of this external knowledge is to improve the user experience of some applications the company has been running already. The main challenges are:

C1. Entity Matching: identify the internal provider information with the external data sources. No matching entities should be discarded.

C2. Context Validation: we have to validate whether the external provider's data belongs to the manufacturing domain.

The use case (U1) is simple: based on the internal metadata of the providers, the company wants to complete their knowledge about them from external sources.

7.4.2 MINTE$^+$ Configuration

To address the challenges of this application and support the use case, MINTE$^+$ was configured with the parameters shown in Table 7.3. As query Q, metadata about the providers, e.g., the provider's name, is sent to MINTE$^+$. As the ontology O, schema.org was configured, in particular, the subset that describes the Organization concept[16] was extended: `theCompany:PartsProvider` (a subclass of `schema:Organization`), having the property `theCompany:industry` with values such as "Semiconductors". Four wrappers were developed for this application. For confidentiality reasons, we can mention just DBpedia and Google Knowledge Graph. To address challenge C1, SILK was configured to provide values of similarity, i.e., it is used in MINTE$^+$ as a similarity function. In this application, only one rule was configured in SILK to measure the similarity between a Google Knowledge Graph molecule with a DBpedia molecule. Only if the organization Wikipedia page[17] in both molecules refer to the same URL, they are considered the same. This is the reason for a threshold of 1.0. DBpedia is selected as major source in the authoritative fusion policy configured for this application. To provide the necessary interface for other systems on top of the MINTE$^+$ API, a new REST method returning just JSON was designed with the company. To address the C2 challenge, a SPARQL Construct query filters the manufacturing context of the molecules (`theCompany:industry = Semiconductors`).

[15] For confidentiality reasons we cannot mention the name.

[16] http://schema.org/Organization

[17] http://schema.org/ContactPage

7.4.3 Results and Lessons Learned

The application is in production state. The company has more than 300 providers in their internal catalog. We evaluated the accuracy of knowledge completion (U1) by randomly selecting 100 molecules and manually creating a gold standard, then compared the results produced by MINTE$^+$ to the gold standard. We obtained 85% accuracy, which means 85 times out of 100 MINTE$^+$ was able to complete the internal knowledge about providers with molecules coming from DBpedia and Google Knowledge Graph. Matching failures are explained mostly by outdated information from the providers, e.g., when the name of a subcontractor has changed. Although the percentage is not high, it still impacts user experience in the company's control system. Thanks to the good results regarding providers, the next step is to apply MINTE$^+$ to other entities handled by the company, such as "Components".

7.5 Summary

In this chapter, we described MINTE$^+$ and discussed its implementation in three domain-specific applications to synthesize RDF molecules into a knowledge graph. The three applications are either under evaluation in the field or in production. The role of semantic web technology is central to the success of the MINTE$^+$ framework. We showed the benefits of the MINTE$^+$ implementation in terms of the configurability and extensibility of its components. The effort to configure, extend, and adapt the MINTE$^+$ implementation is relatively low (new fusion policies, similarity functions, wrappers may be developed and plugged into the framework); state-of-art approaches can be easily integrated. MINTE$^+$ is started to be used in biomedical applications to integrate and transform big data into actionable knowledge. Therefore, MINTE$^+$ is being extended to scale up to large volumes of diverse data.

CHAPTER 8

Conclusions and Future Directions

In this thesis, we studied the problem of retrieving and integrating pieces of knowledge about entities spread over web sources. We proposed a set of strategies and techniques to semantically integrate these pieces of knowledge on-demand. In particular, we tackled the problems of knowledge integration in Chapter 4, entity matching in Chapter 5, knowledge retrieval from heterogeneous web sources in Chapter 6, and we demonstrated the applicability of our methods in real-world domain-specific applications in Chapter 7. In the following sections, we summarize our contributions, discuss main findings and lessons learned, and define future directions for this work from the perspectives of both research and technology.

8.1 Overall contributions and conclusions

The main goal of this thesis is to advance the field of knowledge retrieval and integration by providing a novel set of strategies and techniques to solve the main challenges in a distributed and federated scenario. In this regard, we contributed to answer four research questions. First, we tackled the problem of knowledge integration from heterogeneous sources solving interoperability conflicts at integration time, and we answer the following research question.

> **RQ1**: How can semantics encoded in RDF graphs be exploited to integrate data collected from heterogeneous web sources?

The scenario of knowledge integration from web sources exhibits complex interoperability conflicts, such as domain conflicts, granularity conflicts, and complementary knowledge conflicts. After our literature review presented in Chapter 3, we argue that the state-of-the-art semantic integration frameworks mimic traditional integration approaches, i.e., under the assumption of full access to the datasets, performing heavy ETL pipelines, which is not the case with heterogeneous web sources. To answer **RQ1**, we need a knowledge integration approach that fits better to the scenario of web sources data integration. In consequence, we proposed the MINTE approach, a novel semantic integration approach based on RDF molecules. The MINTE approach is able to integrate semantically equivalent entities from web sources, and it has been designed to exploit the semantics encoded in the data collected from web sources to produce a consolidated knowledge graph. The key characteristics that allow MINTE to exploit semantics are: (a) the use of RDF molecules as the unit of data integration; and (b) the two-fold approach, first by identifying the semantically equivalent entities (using semantic similarity metrics), and then integrating the molecules (using fusion policies). MINTE utilizes RDF molecules in both steps of the integration pipeline, making use of the semantics encoded in the molecules. We empirically demonstrated the advantages of using semantics to integrate data collected from web sources, and we showed the benefits of RDF molecules as the data integration unit. Our theoretical and empirical findings indicate that—in comparison with the state-of-the-art approaches, MINTE integrates heterogeneous data with good accuracy, when interoperability conflicts are present in the web sources, answering research question **RQ1**. Moreover, the MINTE defines a set of configuration parameters making it applicable to a variety of domain-specific applications.

Based on our findings, we contribute to the state-of-the-art in the area of knowledge integration by: 1) defining a new integration approach based on RDF molecules, the approach is tailored for the scenario of web sources integration where the governance of the data stills on the hands of the data producers; 2) formalizing RDF molecules as data integration, and demonstrating its flexibility to deal with different domain-specific applications; and 3) providing a flexible architecture that allows adding state-of-the-art approaches from the Semantic Web community with low effort, increasing the visibility and the impact of new approaches coming from the community. Moreover, MINTE's implementation[1] is open source and accessible to anybody.

The second problem we tackled is determining semantically similar entities over heterogeneous web sources, the obtained results allowed us to answer the following research question.

RQ2: How can semantic similarity metrics facilitate the process of integrating data collected from heterogeneous web sources?

Several approaches have been proposed to compare the similarity between entities, however, the impact of these metrics on the data integration task has not been sufficiently studied. To answer **RQ2**, we first reviewed the state-of-the-art approaches and selected GADES as semantic similarity measure. In order to evaluate different similarity metrics, we defined a semantic similarity framework that includes GADES (a semantic metric) and Jaccard (a non-semantic metric). To perform a fair comparison, both GADES and Jaccard were adapted to work with RDF molecules. We empirically demonstrated that using a similarity metric, i.e., GADES, provides better performance than non-semantic similarity metric, i.e., Jaccard, in the task of integration when the data sources suffer from semantic interoperability problems. The empirical evaluations show the benefits of using semantic similarity approaches to support the problem of integrating pieces of knowledge belonging to the same entity.

Although GADES performed well on the task of determining semantically equivalent entities, it requires a fine-tuning process of its parameters. Moreover, GADES' quality depends on the quality of the ontology defined for the RDF molecules. To avoid the need for a manual fine-tuning intervention, we proposed MateTee a novel similarity metric based on embeddings. We defined a process to produce embeddings from RDF molecules and calculate the distance among these embeddings. As a result, we are able to determine the similarity among entities coming from web sources. We empirically demonstrated the advantages of MateTee, i.e., no manual fine-tuning process is required, and it performs well on knowledge graphs from different domains. To test the accuracy of MateTee, we compared its results with state-of-the-art methods such as GADES, OnSim, as well as state-of-the-art similarity measures available in the CESSM evaluation framework. MateTee exhibited high accuracy and competitive results, even outperforming the results of GADES. This behavior was observed in the collections of proteins

[1] https://github.com/RDF-Molecules/MINTE

for UniProt and the collection of persons from DBpedia. The observed results suggest that representing knowledge encoded in RDF molecules as embeddings provide an accurate method for determining relatedness among entities in knowledge graphs. MateTee's approach won the best paper award at the 17th International Conference on Web Engineering (cf. Appendix B).

We formally and empirically proved that the use of semantic similarity measures improves the task of integrating knowledge from web sources. Based on our findings, we contributed to the state-of-the-art by: 1) demonstrating how semantic interoperability conflicts may be solved by using a semantic similarity metric, i.e., GADES to integrate knowledge web sources; and 2) defining a new similarity metric for RDF molecules based on embeddings, i.e., MateTee. All the source code of the semantic similarity framework is open source[2] and accessible to anybody.

The third problem we tackled in this thesis is building and exploring knowledge graphs on-demand from web sources, answering the following research question:

> **RQ3**: How can knowledge graphs be populated on-demand with data collected from heterogeneous web sources?

Most of the state-of-the-art approaches to build knowledge graphs start with the assumption of full access to datasets, so huge indexes for knowledge exploration can be created. In contrast, web sources provide access just to local views of entities via Web APIs, they are autonomous, independent, evolve on their own pace. To answer question **RQ3**, first, we evaluated the use of these APIs as a door to extract information from different segments of the Web. We devised then FuhSen, an on-demand knowledge retrieval and exploration engine for web sources. We demonstrated how Web APIs, provided by web sources, can we used to create knowledge graphs at query time, integrating the knowledge about entities they contain. The use of RDF molecule wrappers and the MINTE approach facilitate the integration of sources from different segments of the Web. Results of the empirical evaluations suggest that FuhSen is able to effectively integrate pieces of knowledge spread over different web sources on-demand. The experiments suggest that the molecule based integration technique implemented in FuhSen integrates data into a knowledge graph more accurately than existing integration techniques. FuhSen's approach devises a novel knowledge retrieval paradigm incorporating principles of Linked Data and Federated Search engines. FuhSen can be applied in numerous use cases, e.g., related to e-commerce (e.g., price comparison) or human resources management (e.g., build a complete candidate profile from open web data). Moreover, to explore the knowledge graphs built on-demand, we presented FaRBIE, a reactive faceted browsing UI to explore multiple RDF graphs from the LOD Cloud at a time. FaRBIE follows a reactive user interface approach that handles the uncertainty imposed by the intrinsic nature of RDF graphs, with the goal to provide a better user experience. FaRBIE is composed of several reactive UI components which react to changes of the semantics, variations in the size of the data, and the disparities in the response times coming from the different sources, allowing for a real-time user experience. Our experiments suggest that the reactive user interface style used in RDF user interfaces, such as faceted browsing reactivity, is a path for improved overall user experience and becoming a key factor in bringing on-demand built knowledge graphs closer to non-technical users.

We show empirically that FuhSen is able to populate knowledge graphs on-demand from web sources. We also demonstrate the advantages of an on-demand exploration approach, i.e., FaRBIE. Based on our findings, we contribute to the state-of-the-art by: 1) defining a federated

[2] https://github.com/RDF-Molecules

semantic search engine for web sources, which is able to integrate pieces of knowledge about the same entity from different segments of the Web; 2) designing an on-demand approach to build knowledge graphs at query time, which accurately creates a knowledge graph out of web sources; and 3) presenting a novel on-demand exploration paradigm for knowledge graphs, providing a positive user experience to explore entities.

The fourth and final question we answered in the scope of this thesis is the following:

> **RQ4**: How does semantic data integration impact the adaptability of knowledge retrieval systems?

Because of the continuous growth of heterogeneous data on different segments of the Web, new technologies need to be adaptable enough to work on different domain-specific applications. Each domain-specific application contains different degrees of interoperability problems that need to be solved at integration time. To answer **RQ4**, we implemented and integrated all the approaches described throughout this thesis, the result is an open source application named MINTE^{+}. We have applied MINTE^{+} in three different domain-specific applications. For the law enforcement and crime analysis support, we show some of the potential use cases of on-demand knowledge graph creation, i.e., corruption cases, fanaticism, and illegal medication markets. The application developed is compliant with EU regulations and can be used by law enforcement agencies. During the development and evaluation of this application domain, we were able to confirm the feasibility of integrating data from web sources of different segments of the Web, i.e., Deep Web, Social Web, and Dark Web. For the job market analysis, we show how the approach developed in the scope of this thesis is able to produce a complete view of the European job market by integrating web sources. Finally, in the manufacturing domain, we show how the internal knowledge about providers can be completed from open web data sources on-demand.

The integration of a new data source may take 1-2 working days. The parameters defined in MINTE^{+} i.e., the threshold, the similarity function, the fusion policy allowed us to tune the integration approach accordingly to the needs of the specific domain application. We can reuse existing schemata, such as Schema.org, and the DBpedia Ontology. New semantic similarity metrics and fusion policies can be integrated quite easily. Through the successful conclusion of the projects where MINTE^{+} was applied, we are able to answer RQ4 and conclude that using the semantic integration approach MINTE^{+} we can accurately integrate data for the crime investigation domain, for the job market analysis scenario, and for the manufacturing domain. Those are just some of the many domain-specific applications where MINTE^{+} can be used.

8.2 Outlook

In this final section, we describe some possible future directions for this work. In the scope of this thesis, we focused on just some properties of the proposed solutions, e.g., the effectiveness of the MINTE approach. Therefore, there is still room to improve the results proposed in this thesis, e.g., wrt. scalability. Regarding the MINTE integration approach, we envision the following future work:

- Extend the MINTE approach to be context aware; the problem of entity similarity during the integration process has been extensively tackled in the scope of this thesis. However, the context dimension, i.e., two entities are not the same in all contexts [166], could be a new extension to the MINTE framework. A possible addition could be defining a new fusion policy that considers the context of the entities.

- Deep Learning is gaining a lot of attention in all domains including data management [167]. Thus, MINTE integration results could be improved and automated by the usage of Deep Learning models. A possible extension could be the use of reinforcement learning for automatic configuration of MINTE parameters. We can find already research pursuing a similar goal in [168–170].
- In order to improve the information quality in the integration process of large amounts of data, MINTE requires a formal process quality schema. We suggest to follow the criteria defined by Wang et al. [171]: Traceability of the data origin (Data Provenance), Traceability of the loading process (Logging of errors and main events), Referential Integrity (Control on the artificially generated URIs), and Time Variance (Tracking changes over time).

Regarding the semantic similarity framework, it can be extended with the following ideas:

- Add new state-of-the-art similarity metrics for RDF entities based on artificial intelligence approaches. Traverso and Vidal [172] present GARUM, a semantic similarity measure based on machine learning and entity characteristics, a natural step would be analyze and integrate GARUM to the framework.
- The usage of embeddings to solve the problem of entity similarity is a promising research line. We suggest to continue and extend the similarity metric we have defined in this thesis, i.e., MateTee. Possible extensions could be considering not only the explicit knowledge encoded in the RDF molecules but the implicit knowledge as well, i.e., implicit relations can be materialized using inferencing components to then use TransE to create the embeddings. Our intuition tells us the similarity metric accuracy should improve.
- Although MateTee accuracy is comparable with state-of-the-art approaches, it requires a lot of time to produce the embeddings of the entities, making it difficult for real-time scenarios. Another line of research would produce the embeddings on-demand using transfer learning, i.e., based on pre-trained embeddings from open knowledge graphs such as DBpedia, utilize those and train few iterations with the new RDF molecules.

Regarding our federated search engine, in the future, we plan to:

- All improvements and changes made in MINTE and the Similarity Framework discuss in the previous items should be evaluated in FuhSen. These improvements should have a positive impact on FuhSen performance.
- Evolve the concept of RDF molecule even further as the unit of representation for knowledge integration in general, not just for web sources. The definition of an RDF molecule should encode not only the data, but as well as metadata, e.g., provenance information, context information, history of evaluation, and more. The goal is that this additional metadata serves as input to the components in FuhSen to improve the performance at each step.
- Regarding FaRBIE, an interesting line of research would be to study how FaRBIE may foster serendipitous discoveries on-demand on the data coming from web sources. Khalili et al. [173] show how RDF graph exploration allows the discovery of interesting and valuable facts not initially sought for.

In terms of the applicability of the approaches presented in this thesis, we see many opportunities to solve integration problems in the following domains:

- In the healthcare domain where the knowledge of entities is spread over hundreds of IT systems, integrating and exploring these data on-demand may facilitate the analysis of the data produce by healthcare institutions. Aasman et al. [174] present the deployment of a patient knowledge graph for improving patient care and medical research, showing the value of knowledge graphs to provide the information to find patterns in the data and to use those patterns for clinical purposes to improve clinical outcomes.
- The Internet of Things (IoT) is another application domain where the application of the results of this thesis is interesting. The IoT is characterized by the velocity of the generated data, and the use of APIs to access this data. The on-demand knowledge graph creation from the data produced by smart devices in the IoT domain can be in the interest of many companies looking to create value out of this data.

8.3 Closing Remarks

With the increasing amount of data about entities on the Web, the knowledge integration problem is constantly facing new opportunities and challenges. In this thesis, we have shown the benefits of semantic integration approaches to successfully tackle the problem of integrating pieces of knowledge of the same entity spread over web sources. Future research work can build upon the contributions presented in this thesis to devise more flexible and comprehensive integration approaches. Additionally, the pieces of software produced during the development of this thesis are impacting several application domains—resulting in a Fraunhofer IAIS product and taking part in new European research proposals and projects.

Bibliography

[1] D. M. Herzig, P. Mika, R. Blanco and T. Tran, "Federated Entity Search Using On-the-Fly Consolidation", *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, 2013 167, URL: https://doi.org/10.1007/978-3-642-41335-3%5C_11 (cit. on pp. 1, 5, 15, 33, 34).

[2] J. Guo, G. Xu, X. Cheng and H. Li, "Named entity recognition in query", *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, 2009 267 (cit. on p. 1).

[3] J. Pound, P. Mika and H. Zaragoza, "Ad-hoc object retrieval in the web of data", *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, 2010 771, URL: http://doi.acm.org/10.1145/1772690.1772769 (cit. on p. 1).

[4] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi and D. Petrelli, "Hybrid Search: Effectively Combining Keywords and Semantic Searches", *Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications ESWC, Tenerife, Canary Islands, Spain, June 1-5*, 2008 554 (cit. on pp. 2, 34).

[5] R. Usbeck, A. N. Ngomo, L. Bühmann and C. Unger, "HAWK - Hybrid Question Answering Using Linked Data", *Proceedings of the 12th European Semantic Web Conference on the The Semantic Web. Latest Advances and New Domains ESWC, Portoroz, Slovenia, May 31 - June 4*, 2015 353 (cit. on pp. 2, 34).

[6] M.-E. Vidal, K. M. Endris, S. Jozashoori, F. Karim and G. Palma, "Semantic Data Integration of Big Biomedical Data for Supporting of Personalised Medicine", (in press) (cit. on pp. 3, 4, 23, 24, 30).

[7] I. Traverso, M.-E. Vidal, B. Kämpgen and Y. Sure-Vetter, "GADES: A Graph-based Semantic Similarity Measure", *Proceedings of the 12th International Conference on Semantic Systems, SEMANTICS 2016, September 12-15, Leipzig, Germany*, ACM, 2016 101 (cit. on pp. 5, 7, 48, 53, 54, 56, 61, 76, 77, 95, 98, 102).

[8] J. D. Fernandez, A. Llaves and O. Corcho, "Efficient RDF Interchange (ERI) Format for RDF Data Streams", *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*, 2014 244 (cit. on pp. 5, 18, 76).

[9] T. Mikolov, K. Chen, G. Corrado and J. Dean,
Efficient Estimation of Word Representations in Vector Space,
CoRR **abs/1301.3781** (2013), arXiv: 1301.3781,
URL: http://arxiv.org/abs/1301.3781 (cit. on pp. 6, 32, 33).

[10] S. Brin and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, Proccedings of the 7th International World Wide Web Conference WWW, Brisbane, Australia, April 14-18 **30** (1998) 107 (cit. on pp. 11, 33, 74).

[11] C. Bizer, T. Heath and T. Berners-Lee, *Linked Data - The Story So Far*,
Int. J. Semantic Web Inf. Syst. **5** (2009) 1 (cit. on p. 12).

[12] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak and Z. G. Ives,
“DBpedia: A Nucleus for a Web of Open Data”, *Proceedings of the 6th International Semantic Web Conference ISWC + ASWC, Busan, Korea, November 11-15*, 2007 722 (cit. on p. 12).

[13] J. G. Breslin, A. Passant and S. Decker, *The social semantic web*, Springer, 2009,
ISBN: 9783642011719 (cit. on p. 13).

[14] J. Devine and F. Egger-Sider, *Beyond google: the invisible web in the academic library*, The Journal of Academic Librarianship **30** (2004) 265 (cit. on p. 13).

[15] B. He, M. Patel, Z. Zhang and K. C. Chang, *Accessing the deep web*,
Commun. ACM **50** (2007) 94 (cit. on p. 13).

[16] C. Mattmann, *Search of the Deep and Dark Web via DARPA Memex*,
2015 AGU Fall Meeting (2015) (cit. on p. 13).

[17] M. Shokouhi and L. Si, *Federated Search*,
Foundations and Trends in Information Retrieval (2011) (cit. on p. 15).

[18] S. Elbassuoni and R. Blanco, “Keyword search over RDF graphs”,
Proceedings of the 20th Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011 237 (cit. on p. 15).

[19] T. Tran, H. Wang, S. Rudolph and P. Cimiano, “Top-k Exploration of Query Candidates for Efficient Keyword Search on Graph-Shaped (RDF) Data”,
Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 - April 2, Shanghai, China, 2009 405,
URL: https://doi.org/10.1109/ICDE.2009.119 (cit. on p. 15).

[20] A. Doan, A. Y. Halevy and Z. G. Ives, *Principles of Data Integration*,
Morgan Kaufmann, 2012, ISBN: 978-0-12-416044-6 (cit. on pp. 15, 22, 26).

[21] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes,
S. Hellmann, M. Morsey, P. van Kleef, S. Auer and C. Bizer,
DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia,
Semantic Web **6** (2015) 167, URL: https://doi.org/10.3233/SW-140134
(cit. on p. 16).

[22] D. Vrandecic and M. Krötzsch, *Wikidata: a free collaborative knowledgebase*,
Commun. ACM **57** (2014) 78, URL: http://doi.acm.org/10.1145/2629489
(cit. on p. 16).

[23] R. Blanco, B. B. Cambazoglu, P. Mika and N. Torzec,
"Entity Recommendations in Web Search",
The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II, 2013 33,
URL: https://doi.org/10.1007/978-3-642-41338-4%5C_3 (cit. on p. 16).

[24] H. Paulheim,
Knowledge graph refinement: A survey of approaches and evaluation methods,
Semantic Web **8** (2017) 489, URL: https://doi.org/10.3233/SW-160218
(cit. on p. 16).

[25] S. Auer, *Enterprise Knowledge Graphs*,
URL: https://de.slideshare.net/semanticsconference/sren-auer-enterprise-knowledge-graphs (cit. on p. 17).

[26] A. Bernstein, J. A. Hendler and N. F. Noy, *A new look at the semantic web*,
Communications of the ACM **59** (2016) 35 (cit. on pp. 16, 54).

[27] M. Arenas, C. Gutierrez and J. Perez, "Foundations of RDF Databases",
Proceedings of the 5th International Summer School 2009, Brixen-Bressanone, Italy, August 30 - September 4, Tutorial Lectures, 2009 158 (cit. on pp. 16, 76).

[28] J. Perez, M. Arenas and C. Gutierrez, *Semantics and complexity of SPARQL*,
ACM Trans. Database Syst. **34** (2009) 16:1 (cit. on p. 17).

[29] R. Angles and C. Gutierrez, "The Expressive Power of SPARQL", *Proceedings of the 7th International Semantic Web Conference, ISWC, Karlsruhe, Germany, October 26-30*, 2008 114 (cit. on p. 18).

[30] M. Brambilla and S. Ceri,
Designing exploratory search applications upon web data sources, Springer, 2012 61
(cit. on p. 19).

[31] R. W. White, B. Kules and B. B. Bederson,
Exploratory search interfaces: categorization, clustering and beyond: report on the XSI 2005 workshop at the Human-Computer Interaction Laboratory, University of Maryland,
SIGIR Forum **39** (2005) 52 (cit. on p. 19).

[32] F. Haag, S. Lohmann, S. Siek and T. Ertl,
"Visual Querying of Linked Data with QueryVOWL",
Joint Proceedings of the 1st International Workshop on Summarizing and Presenting Entities and Ontologies and the 3rd International Workshop on Human Semantic Web Interfaces (SumPre 2015, HSWI 2015) co-located with the 12th Extended Semantic Web Conference (ESWC 2015), Portoroz, Slovenia, June 1, 2015. 2015 (cit. on p. 20).

[33] Q. Zhou, C. Wang, M. Xiong, H. Wang and Y. Yu,
"SPARK: Adapting Keyword Query to Semantic Search",
The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. 2007 694 (cit. on p. 20).

[34] E. Franconi, P. Guagliardo, M. Trevisan and S. Tessaris,
"Quelo: an Ontology-Driven Query Interface", *Proceedings of the 24th International Workshop on Description Logics (DL 2011), Barcelona, Spain, July 13-16, 2011*, 2011
(cit. on p. 20).

[35] M. Arenas, B. C. Grau, E. Kharlamov, S. Marciuska and D. Zheleznyakov, *Faceted search over RDF-based knowledge graphs*, Journal of Web Semantics **37** (2016) (cit. on pp. 20, 75, 82, 103).

[36] D. Tunkelang, *Faceted Search*, Synthesis Lectures on Information Concepts, Retrieval, and Services, Morgan and Claypool Publishers, 2009 (cit. on p. 20).

[37] T. Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, E. Prudhommeaux and M. M. C. Schraefel, "Tabulator Redux: Browsing and Writing Linked Data", *Proceedings of the WWW2008 Workshop on Linked Data on the Web, LDOW 2008, Beijing, China, April 22.* 2008 (cit. on p. 20).

[38] P. Heim, J. Ziegler and S. Lohmann, "gFacet: A Browser for the Web of Data", *Proceedings of the International Workshop on Interacting with Multimedia Content in the Social Semantic Web (IMC-SSW 08) Koblenz, Germany, December 3.* 2008 (cit. on p. 20).

[39] M. Hildebrand, J. van Ossenbruggen and L. Hardman, "facet: A Browser for Heterogeneous Semantic Web Repositories", *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings*, 2006 272 (cit. on p. 20).

[40] H. Bast, F. Bäurle, B. Buchhold and E. Haussmann, "Easy access to the freebase dataset", *23rd International World Wide Web Conference, WWW 14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, 2014 95 (cit. on p. 20).

[41] M. Lenzerini, "Data Integration: A Theoretical Perspective", *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA*, 2002 233 (cit. on pp. 20, 21).

[42] G. Wiederhold, *Mediators in the Architecture of Future Information Systems*, IEEE Computer **25** (1992) 38, URL: `https://doi.org/10.1109/2.121508` (cit. on p. 21).

[43] A. Y. Halevy, *Answering queries using views: A survey*, VLDB J. **10** (2001) 270 (cit. on p. 21).

[44] A. Y. Levy, A. O. Mendelzon, Y. Sagiv and D. Srivastava, "Answering Queries Using Views", *Proceedings of the 14th ACM Symposium on Principles of Database Systems SIGACT-SIGMOD-SIGART, May 22-25, 1995, San Jose, California, USA*, 1995 95 (cit. on p. 21).

[45] S. Bergamaschi, E. Domnori, F. Guerra, M. Orsini, R. T. Lado and Y. Velegrakis, *Keymantic: Semantic Keyword-based Searching in Data Integration Systems*, PVLDB **3** (2010) 1637 (cit. on p. 22).

[46] P. Frischmuth, J. Klimek, S. Auer, S. Tramp, J. Unbehauen, K. Holzweissig and C.-M. Marquardt, *Linked Data in Enterprise Information Integration*, Semantic Web (2012) 1 (cit. on pp. 22, 28).

[47] G. F. Ashby and D. M. Ennis, *Similarity measures*, Scholarpedia **2** (2007) 4116 (cit. on p. 23).

[48] L. Belanche and J. Orozco, “Things to Know about a (dis)similarity Measure”, *Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems KES, September 12-14, Kaiserslautern, Germany*, 2011 100 (cit. on p. 23).

[49] R. Bellazzi, *Big data and biomedical informatics: a challenging opportunity.*, Yearb Med Inform **9** (2014) 8 (cit. on p. 23).

[50] N. F. Noy, *Semantic Integration: A Survey Of Ontology-Based Approaches*, SIGMOD Record **33** (2004) 65 (cit. on p. 26).

[51] D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, R. Rosati and G. A. Ruberti, *Ontology-Based Data Access and Integration*, Encyclopedia of Database Systems (2017) 1 (cit. on p. 26).

[52] J. F. Sequeda and D. P. Miranker, *A Pay-As-You-Go Methodology for Ontology-Based Data Access*, IEEE Internet Computing **21** (2017) 92 (cit. on p. 26).

[53] J. F. Sequeda, M. Arenas and D. P. Miranker, “OBDA: Query Rewriting or Materialization? In Practice, Both!”, *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, 2014 535, URL: https://doi.org/10.1007/978-3-319-11964-9%5C_34 (cit. on p. 26).

[54] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro and G. Xiao, *Ontop: Answering SPARQL queries over relational databases*, Semantic Web **8** (2017) 471, URL: https://doi.org/10.3233/SW-160217 (cit. on pp. 26, 29, 30, 33).

[55] T. Bagosi, D. Calvanese, J. Hardi, S. Komla-Ebri, D. Lanti, M. Rezk, M. Rodriguez-Muro, M. Slusnys and G. Xiao, “The Ontop Framework for Ontology Based Data Access”, *The Semantic Web and Web Science - 8th Chinese Conference, CSWS 2014, Wuhan, China, August 8-12, 2014, Revised Selected Papers*, 2014 67, URL: https://doi.org/10.1007/978-3-662-45495-4%5C_6 (cit. on pp. 26, 29, 30, 33).

[56] F. Priyatna, O. Corcho and J. F. Sequeda, “Formalisation and experiences of R2RML-based SPARQL to SQL query translation using morph”, *23rd International World Wide Web Conference, WWW ’14, Seoul, Republic of Korea, April 7-11, 2014*, 2014 479, URL: http://doi.acm.org/10.1145/2566486.2567981 (cit. on p. 26).

[57] A. C. Junior, C. Debruyne and D. O’Sullivan, “Juma: An Editor that Uses a Block Metaphor to Facilitate the Creation and Editing of R2RML Mappings”, *The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portoroz, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, 2017 87, URL: https://doi.org/10.1007/978-3-319-70407-4%5C_17 (cit. on p. 26).

[58] J. Bak, M. Blinkiewicz and A. Lawrynowicz, "User-friendly Visual Creation of R2RML Mappings in SQuaRE", *Proceedings of the Third International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 22*, 2017 139, URL: http://ceur-ws.org/Vol-1947/paper13.pdf (cit. on p. 26).

[59] J. Bak and M. Blinkiewicz, "SQuaRE: A Visual Tool For Creating R2RML Mappings", *Proceedings of the ISWC 2016 Posters and Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19*, 2016, URL: http://ceur-ws.org/Vol-1690/paper62.pdf (cit. on p. 26).

[60] J. F. Sequeda and D. P. Miranker, *Ultrawrap: SPARQL execution on relational data*, J. Web Sem. **22** (2013) 19, URL: https://doi.org/10.1016/j.websem.2013.08.002 (cit. on pp. 27, 29, 30, 33).

[61] J. F. Sequeda and D. P. Miranker, "Ultrawrap Mapper: A Semi-Automatic Relational Database to RDF (RDB2RDF) Mapping Tool", *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015.* 2015, URL: http://ceur-ws.org/Vol-1486/paper%5C_105.pdf (cit. on pp. 27, 29, 30, 33).

[62] F. Priyatna, R. Alonso-Calvo, S. Paraiso-Medina, G. Padron-Sanchez and O. Corcho, "R2RML-based Access and Querying to Relational Clinical Data with Morph-RDB", *Proceedings of the 8th Semantic Web Applications and Tools for Life Sciences International Conference, Cambridge UK, December 7-10, 2015.* 2015 142, URL: http://ceur-ws.org/Vol-1546/paper%5C_21.pdf (cit. on p. 27).

[63] S. Geisler, C. Quix, A. Schmeink and D. Kensche, "Ontology-based Data Integration: A Case Study in Clinical Trials", *Database Technology For Life Sciences And Medicine*, World Scientific, 2010 115 (cit. on p. 27).

[64] R. Piro, Y. Nenov, B. Motik, I. Horrocks, P. Hendler, S. Kimberly and M. Rossman, "Semantic Technologies for Data Analysis in Health Care", *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, 2016 400, URL: https://doi.org/10.1007/978-3-319-46547-0%5C_34 (cit. on p. 27).

[65] N. Freire, V. Charles and A. Isaac, "Evaluation of Schema.org for Aggregation of Cultural Heritage Metadata", *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, 2018 225, URL: https://doi.org/10.1007/978-3-319-93417-4%5C_15 (cit. on p. 27).

[66] C. A. Knoblock, P. A. Szekely, E. E. Fink, D. Degler, D. Newbury, R. Sanderson, K. Blanch, S. Snyder, N. Chheda, N. Jain, R. R. Krishna, N. B. Sreekanth and Y. Yao, "Lessons Learned in Building Linked Data for the American Art Collaborative", *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part II*, 2017 263, URL: https://doi.org/10.1007/978-3-319-68204-4%5C_26 (cit. on p. 27).

[67] D. A. Ostrowski and M. Kim,
"A Semantic Based Framework for the Purpose of Big Data Integration",
11th IEEE International Conference on Semantic Computing, ICSC 2017, San Diego, CA, USA, January 30 - February 1, 2017, 2017 305,
URL: https://doi.org/10.1109/ICSC.2017.62 (cit. on p. 27).

[68] C. A. Knoblock and P. A. Szekely, *Exploiting Semantics for Big Data Integration*,
AI Magazine **36** (2015) 25,
URL: http://www.aaai.org/ojs/index.php/aimagazine/article/view/2565
(cit. on p. 27).

[69] G. Xiao, D. Hovland, D. Bilidas, M. Rezk, M. Giese and D. Calvanese,
"Efficient Ontology-Based Data Integration with Canonical IRIs",
The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings, 2018 697,
URL: https://doi.org/10.1007/978-3-319-93417-4%5C_45 (cit. on p. 27).

[70] C. Sutanay, A. Khushbu, P. Sumit, Z. Baichuan, P. Meg, S. Will and T. Mathew,
NOUS: Construction and Querying of Dynamic Knowledge Graphs,
arXiv preprint arXiv:1606.02314 (2016) (cit. on p. 27).

[71] P. Thomas, A. Youssef, K. Juhana and C. Re,
"Wikipedia Knowledge Graph with DeepDive", *Proccedings of the 10th International AAAI Conference on Web and Social Media, Köln, Germany, May 17–20*, 2016
(cit. on p. 27).

[72] D. Xin, G. Evgeniy, H. Geremy, H. Wilko, L. Ni, M. Kevin, S. Thomas, S. Shaohua and Z. Wei, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion",
Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, August 24-27, ACM, 2014 601
(cit. on p. 27).

[73] K. D. Bollacker, C. Evans, P. Paritosh, T. Sturge and J. Taylor,
"Freebase: a collaboratively created graph database for structuring human knowledge",
Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008, 2008 1247,
URL: http://doi.acm.org/10.1145/1376616.1376746 (cit. on p. 28).

[74] C. A. Knoblock, P. A. Szekely, J. L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyan and P. Mallick,
"Semi-automatically Mapping Structured Sources into the Semantic Web",
Proceedings of the 9th Extended Semantic Web Conference ESWC, May 27-31, Heraklion, Crete, Greece, 2012 375 (cit. on pp. 28, 30).

[75] P. A. Szekely, C. A. Knoblock, F. Yang, X. Zhu, E. E. Fink, R. Allen and G. Goodlander,
"Connecting the Smithsonian American Art Museum to the Linked Data Cloud",
The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings, 2013 593,
URL: https://doi.org/10.1007/978-3-642-38288-8%5C_40 (cit. on p. 28).

[76] P. A. Szekely, C. A. Knoblock and J. Slepicka, "Building and Using a Knowledge Graph to Combat Human Trafficking", *Proccedings of the 14th International Semantic Web Conference, The Semantic Web ISWC, Bethlehem, PA, USA, October 11-15*, 2015 205 (cit. on pp. 28, 34, 73).

[77] S. Andreas, M. Andrea, I. Robert, M. P. N, B. Christian and B. Christian, "LDIF - A framework for large-scale Linked Data integration", *Proceedings of the 21st International World Wide Web Conference WWW, Developers Track, Lyon, France, April 16-20*, 2012 (cit. on pp. 28, 30, 33).

[78] R. Isele and C. Bizer, *Active learning of expressive linkage rules using genetic programming*, Journal of Web Semantics **23** (2013) 2 (cit. on pp. 28, 30, 39, 105, 106).

[79] J. Michelfeit and T. Knap, "Linked Data Fusion in ODCleanStore", *Proceedings of the ISWC 2012 Posters and Demonstrations Track, November 11-15, Boston, USA*, 2012 (cit. on pp. 28, 30, 33, 39).

[80] T. Knap, M. Kukhar, B. Machac, P. Skoda, J. Tomes and J. Vojt, "UnifiedViews: An ETL Framework for Sustainable RDF Data Processing", *The Semantic Web: ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, Revised Selected Papers*, 2014 379 (cit. on p. 28).

[81] T. Knap, P. Skoda, J. Klimek and M. Necasky, "UnifiedViews: Towards ETL Tool for Simple yet Powerfull RDF Data Management", *Proceedings of the Dateso Annual International Workshop on DAtabases, TExts, Specifications and Objects, Neprivec u Sobotky, Jicin, Czech Republic, April 14, 2015*, 2015 111 (cit. on p. 28).

[82] J. Michelfeit, T. Knap and M. Necasky, *Linked Data Integration with Conflicts*, CoRR **abs/1410.7990** (2014), URL: `http://arxiv.org/abs/1410.7990` (cit. on p. 29).

[83] Y. Tzitzikas, N. Minadakis, Y. Marketakis, P. Fafalios, C. Allocca, M. Mountantonakis and I. Zidianaki, "MatWare : Constructing and Exploiting Domain Specific Warehouses by Aggregating Semantic Data", *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, 2014 721, URL: `https://doi.org/10.1007/978-3-319-07443-6%5C_48` (cit. on pp. 29, 30, 33).

[84] P. N. Mendes, H. Mühleisen and C. Bizer, "Sieve: linked data quality assessment and fusion", *Proceedings of the 2012 Joint EDBT/ICDT Workshops, Berlin, Germany, March 30, 2012*, 2012 116 (cit. on pp. 29, 39).

[85] L. Ding, T. Finin, A. Joshi, Y. Peng, P. P. da Silva and D. McGuinness, "Tracking RDF Graph Provenance using RDF Molecules", *Proceedings of the 4th International Semantic Web Conference ISWC, Galway, Ireland, November 6-10*, 2005 156 (cit. on p. 29).

[86] M. Acosta, M. Vidal, T. Lampo, J. Castillo and E. Ruckhaus, "ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints", *Proceedings of the 10th International Conference on The Semantic Web ISWC, Bonn, Germany, October 23-27*, 2011 18 (cit. on pp. 29, 30, 33, 38).

[87] A. Schwarte, P. Haase, K. Hose, R. Schenkel and M. Schmidt, "FedX: Optimization Techniques for Federated Query Processing on Linked Data", *Proceedings of the 10th International Conference on The Semantic Web ISWC, Bonn, Germany, October 23-27*, 2011 601 (cit. on pp. 29, 30, 38).

[88] O. Görlitz and S. Staab, "SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions", *Proceedings of the 2nd International Workshop on Consuming Linked Data (COLD2011), October 23, Bonn, Germany*, 2011 (cit. on pp. 29, 30, 38).

[89] K. Alexander, R. Cyganiak, M. Hausenblas and J. Zhao, *Describing Linked Datasets On the Design and Usage of voiD, the Vocabulary Of Interlinked Datasets*, In Proceedings of the Linked Data on the Web Workshop, Madrid, Spain (2009) (cit. on p. 29).

[90] M. Saleem, S. S. Padmanabhuni, A. N. Ngomo, A. Iqbal, J. S. Almeida, S. Decker and H. F. Deus, *TopFed: TCGA Tailored Federated Query Processing and Linking to LOD*, J. Biomedical Semantics **5** (2014) 47, URL: `https://doi.org/10.1186/2041-1480-5-47` (cit. on pp. 29, 30).

[91] M. A. Jaro, *Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida*, Journal of the American Statistical Association **84** (1989) 414 (cit. on p. 30).

[92] W. E. Winkler, *String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.*, In Proceedings ofthe Section on Survey Research. Washington, DC (1990) 354 (cit. on p. 30).

[93] A. N. Ngomo and S. Auer, "LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data", *IJCAI 2011*, 2011 2312 (cit. on pp. 30, 39).

[94] S. Hillner and A. N. Ngomo, "Parallelizing LIMES for large-scale link discovery", *Proceedings the 7th International Conference on Semantic Systems, I-SEMANTICS 2011, Graz, Austria, September 7-9, 2011*, 2011 9, URL: `http://doi.acm.org/10.1145/2063518.2063520` (cit. on p. 31).

[95] F. M. Suchanek, S. Abiteboul and P. Senellart, *PARIS: Probabilistic Alignment of Relations, Instances, and Schema*, PVLDB **5** (2011) 157, URL: `http://www.vldb.org/pvldb/vol5/p157%5C_fabianmsuchanek%5C_vldb2012.pdf` (cit. on pp. 31, 33).

[96] J. Urbani, S. Kotoulas, J. Maassen, F. van Harmelen and H. E. Bal, *WebPIE: A Web-scale Parallel Inference Engine using MapReduce*, J. Web Sem. **10** (2012) 59, URL: `https://doi.org/10.1016/j.websem.2011.05.004` (cit. on p. 31).

[97] G. Jeh and J. Widom, "SimRank: a measure of structural-context similarity", *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, Edmonton, Alberta, Canada*, 2002 538, URL: `http://doi.acm.org/10.1145/775047.775126` (cit. on p. 31).

[98] C. Böhm, G. de Melo, F. Naumann and G. Weikum, "LINDA: distributed web-of-data-scale entity matching", *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02*, 2012 2104, URL: http://doi.acm.org/10.1145/2396761.2398582 (cit. on p. 31).

[99] C. Paul, A. Rettinger, A. Mogadala, C. A. Knoblock and P. A. Szekely, "Efficient Graph-Based Document Similarity", *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, Proceedings*, 2016 334, URL: https://doi.org/10.1007/978-3-319-34129-3%5C_21 (cit. on p. 31).

[100] V. Efthymiou, K. Stefanidis and V. Christophides, "Minoan ER: Progressive Entity Resolution in the Web of Data", *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16.* 2016 670, URL: https://doi.org/10.5441/002/edbt.2016.79 (cit. on p. 31).

[101] A. Grover and J. Leskovec, "node2vec: Scalable Feature Learning for Networks", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 13-17, San Francisco, CA, USA*, 2016 855 (cit. on pp. 32, 33).

[102] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data", *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., 2013 2787 (cit. on pp. 32, 33, 55, 60, 61).

[103] A. Bordes, J. Weston, R. Collobert and Y. Bengio, "Learning Structured Embeddings of Knowledge Bases", *Proceedings of the 25th Conference on Artificial Intelligence AAAI, August 7-11, San Francisco, California, USA*, 2011 (cit. on p. 32).

[104] P. Ristoski and H. Paulheim, "RDF2Vec: RDF Graph Embeddings for Data Mining", *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, Proceedings, Part I*, 2016 498, URL: https://doi.org/10.1007/978-3-319-46523-4%5C_30 (cit. on pp. 32, 33).

[105] L. Bai, L. Rossi, L. Cui and E. R. Hancock, "A transitive aligned Weisfeiler-Lehman subtree kernel", *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, 2016 396, URL: https://doi.org/10.1109/ICPR.2016.7899666 (cit. on p. 32).

[106] M. Cochez, P. Ristoski, S. P. Ponzetto and H. Paulheim, "Global RDF Vector Space Embeddings", *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, 2017 190, URL: https://doi.org/10.1007/978-3-319-68288-4%5C_12 (cit. on p. 33).

[107] J. Pennington, R. Socher and C. D. Manning, "Glove: Global Vectors for Word Representation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014 1532, URL: http://aclweb.org/anthology/D/D14/D14-1162.pdf (cit. on p. 33).

[108] P. Folz, G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, "SemLAV: Querying Deep Web and Linked Open Data with SPARQL", *The Semantic Web: ESWC 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers*, 2014 332, URL: https://doi.org/10.1007/978-3-319-11955-7%5C_44 (cit. on p. 34).

[109] G. Montoya, L. D. Ibanez, H. Skaf-Molli, P. Molli and M. Vidal, *SemLAV: Local-As-View Mediation for SPARQL Queries*, Trans. Large-Scale Data- and Knowledge-Centered Systems **13** (2014) 33, URL: https://doi.org/10.1007/978-3-642-54426-2%5C_2 (cit. on p. 34).

[110] M. Huber, *Social Snapshot Framework: Crime Investigation on Online Social Networks*, ERCIM News **2012** (2012) (cit. on p. 34).

[111] W. Hu, H. Qiu, J. Huang and M. Dumontier, *BioSearch: a semantic search engine for Bio2RDF*, Database **2017** (2017) bax059 (cit. on p. 35).

[112] M. Dumontier, C. J. O. Baker, J. Baran, A. Callahan, L. L. Chepelev, J. Cruz-Toledo, N. R. D. Rio, G. Duck, L. I. Furlong, N. Keath, D. Klassen, J. P. McCusker, N. Queralt-Rosinach, M. Samwald, N. Villanueva-Rosales, M. D. Wilkinson and R. Hoehndorf, *The Semanticscience Integrated Ontology (SIO) for biomedical research and knowledge discovery*, J. Biomedical Semantics **5** (2014) 14, URL: https://doi.org/10.1186/2041-1480-5-14 (cit. on p. 35).

[113] M. Arenas, B. C. Grau, E. Kharlamov, S. Marciuska and D. Zheleznyakov, "Towards semantic faceted search", *23rd International World Wide Web Conference, WWW 14, Seoul, Republic of Korea, April 7-11, Companion Volume*, ACM, 2014 219 (cit. on pp. 35, 74, 89).

[114] M. Arenas, B. C. Grau, E. Kharlamov, S. Marciuska, D. Zheleznyakov and E. Jimenez-Ruiz, "SemFacet: semantic faceted search over yago", *23rd International World Wide Web Conference, WWW 14, Seoul, Republic of Korea, April 7-11, Companion Volume*, ACM, 2014 123 (cit. on pp. 35, 89).

[115] M. Arenas, B. C. Grau, E. Kharlamov, S. Marciuska and D. Zheleznyakov, "Enabling Faceted Search over OWL 2 with SemFacet", *Proceedings of the 11th International Workshop on OWL: Experiences and Directions (OWLED 2014) co-located with 13th International Semantic Web Conference on (ISWC 2014), Riva del Garda, Italy, October 17-18*, CEUR-WS, 2014 121 (cit. on pp. 35, 74).

[116] C. Stadler, M. Martin and S. Auer, "Exploring the web of spatial data with facete", *23rd International World Wide Web Conference, WWW 14, Seoul, Republic of Korea, April 7-11, Companion Volume*, ACM, 2014 175 (cit. on pp. 35, 74, 75).

[117] S. Ferre,
"Expressive and Scalable Query-Based Faceted Search over SPARQL Endpoints",
The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, Part II, vol. 8797, Lecture Notes in Computer Science, Springer, 2014 438 (cit. on p. 35).

[118] A. Khalili, A. Loizou and F. van Harmelen, "Adaptive Linked Data-Driven Web Components: Building Flexible and Reusable Semantic Web Interfaces",
13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, Proceedings, vol. 9678, Lecture Notes in Computer Science, Springer, 2016 677 (cit. on pp. 36, 74, 86, 89).

[119] A. Khalili, P. V. den Besselaar and K. A. de Graaf,
"FERASAT: A Serendipity-Fostering Faceted Browser for Linked Data",
The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, Proceedings, 2018 351,
URL: https://doi.org/10.1007/978-3-319-93417-4%5C_23 (cit. on p. 36).

[120] A. Stolz and M. Hepp,
"Adaptive Faceted Search for Product Comparison on the Web of Data",
Engineering the Web in the Big Data Era - 15th International Conference, ICWE 2015, Rotterdam, The Netherlands, June 23-26, vol. 9114, Lecture Notes in Computer Science, Springer, 2015 420 (cit. on p. 36).

[121] D. Collarana, M. Galkin, I. T. Ribon, M. Vidal, C. Lange and S. Auer,
"MINTE: semantically integrating RDF graphs",
Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics WIMS, Amantea, Italy, June 19-22, 2017 22:1,
URL: http://doi.acm.org/10.1145/3102254.3102280 (cit. on pp. 37, 95, 100).

[122] D. Collarana, M. Galkin, I. T. Ribon, C. Lange, M. Vidal and S. Auer,
"Semantic Data Integration for Knowledge Graph Construction at Query Time",
11th IEEE International Conference on Semantic Computing ICSC, San Diego, CA, USA, January 30 - February 1, 2017 109,
URL: https://doi.org/10.1109/ICSC.2017.85 (cit. on pp. 37, 53, 71).

[123] M. Galkin, D. Collarana, I. T. Ribon, M. Vidal and S. Auer,
"SJoin: A Semantic Join Operator to Integrate Heterogeneous RDF Graphs",
Database and Expert Systems Applications - 28th International Conference DEXA, Lyon, France, August 28-31, Proceedings, Part I, 2017 206,
URL: https://doi.org/10.1007/978-3-319-64468-4%5C_16 (cit. on pp. 37, 71).

[124] J. Munkres, *Algorithms for the assignment and transportation problems*,
Journal of the society for industrial and applied mathematics **5** (1957) 32 (cit. on p. 47).

[125] B. N. Grosof, I. Horrocks, R. Volz and S. Decker,
"Description logic programs: combining logic programs with description logic",
Proceedings of the 12th International World Wide Web Conference, WWW, May 20-24, Budapest, Hungary, 2003 48 (cit. on p. 48).

[126] P. Hitzler, M. Krötzsch and S. Rudolph, *Foundations of Semantic Web Technologies*,
Chapman and Hall/CRC Press, 2010 (cit. on p. 48).

[127] D. Collarana, M. Galkin, C. Lange, I. Grangel-Gonzalez, M. Vidal and S. Auer, "FuhSen: A Federated Hybrid Search Engine for Building a Knowledge Graph On-Demand (Short Paper)", *On the Move to Meaningful Internet Systems: OTM Conferences - Confederated International Conferences: CoopIS, CTC, and ODBASE, October 24-28, Rhodes, Greece, Proceedings*, 2016 752 (cit. on pp. 48, 65, 71).

[128] C. Paul, A. Rettinger, A. Mogadala, C. A. Knoblock and P. Szekely, "Efficient Graph-Based Document Similarity", *The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC, Heraklion, Crete, Greece, May 29 - June 2, Proceedings*, 2016 334 (cit. on p. 48).

[129] C. Morales, D. Collarana, M. Vidal and S. Auer, "MateTee: A Semantic Similarity Metric Based on Translation Embeddings for Knowledge Graphs", *Web Engineering - 17th International Conference, ICWE 2017, Rome, Italy, June 5-8, Proceedings*, 2017 246, URL: https://doi.org/10.1007/978-3-319-60131-1%5C_14 (cit. on p. 53).

[130] S. Lam and C. Hayes, "Using the structure of DBpedia for exploratory search", *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, 2013 (cit. on p. 56).

[131] J. Benik, C. Chang, L. Raschid, M.-E. Vidal, G. Palma and A. Thor, "Finding Cross Genome Patterns in Annotation Graphs", *Data Integration in the Life Sciences: 8th International Conference, DILS, College Park, MD, USA, June 28-29. Proceedings*, Springer Berlin Heidelberg, 2012 21, ISBN: 978-3-642-31040-9 (cit. on pp. 56, 67, 68).

[132] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks", *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics AISTATS, May 13-15, Chia Laguna Resort, Sardinia, Italy*, 2010 249 (cit. on p. 60).

[133] C. Pesquita, D. Pessoa, D. Faria and F. Couto, *CESSM: Collaborative evaluation of semantic similarity measures*, JB2009: Challenges in Bioinformatics **157** (2009) 190 (cit. on p. 65).

[134] D. Devos and A. Valencia, *Practical limits of function prediction*, Proteins: Structure, Function, and Bioinformatics **41** (2000) 98 (cit. on p. 65).

[135] C. Pesquita, D. Pessoa, D. Faria and F. Couto, *CESSM: Collaborative evaluation of semantic similarity measures*, JB2009: Challenges in Bioinformatics **157** (2009) (cit. on p. 65).

[136] T. Smith and M. Waterman, *Identification of common molecular subsequences*, Journal of Molecular Biology **147** (1981) (cit. on p. 65).

[137] P. Resnik, *Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language*, Journal Of Artificial Intelligence Research **11** (1999) 95 (cit. on p. 66).

[138] D. Lin, "An information-theoretic definition of similarity", *ICML*, vol. 98, 1998 (cit. on p. 66).

[139] P. Lord, R. Stevens, A. Brass and C. Goble, *Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation*, Bioinformatics **19** (2003) 1275 (cit. on p. 66).

[140] J. L. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. M. Mato, L. A. Martinez-Cruz, F. J. Corrales and A. Rubio,
Correlation between Gene Expression and GO Semantic Similarity,
IEEE/ACM Trans. Comput. Biology Bioinformatics **2** (2005) 330 (cit. on p. 66).

[141] F. M. Couto, M. J. Silva and P. Coutinho,
Measuring semantic similarity between Gene Ontology terms,
Data Knowl. Eng. **61** (2007) 137 (cit. on p. 66).

[142] C. Pesquita, D. Faria, H. Bastos, A. O. Falcao and F. M. Couto,
"Evaluating GO-based semantic similarity measures",
Proceedings of the 10th annual Bio-Ontologies Meeting BIOONTOLOGIES, 2007 37 (cit. on pp. 66, 68).

[143] V. Pekar and S. Staab, "Taxonomy learning: factoring the structure of a taxonomy into a semantic classification decision", *COLING 02 Proceedings of the 19th international conference on Computational linguistics*, vol. 2,
Association for Computational Linguistics, 2002 1 (cit. on pp. 67, 68).

[144] I. T. Ribon, M. Vidal and G. Palma,
"OnSim: A Similarity Measure for Determining Relatedness Between Ontology Terms",
Data Integration in the Life Sciences - 11th International Conference, DILS 2015, Los Angeles, CA, USA, July 9-10, Proceedings, 2015 70 (cit. on pp. 67, 68).

[145] I. T. Ribon and M. Vidal, "Exploiting information content and semantics to accurately compute similarity of GO-based annotated entities",
IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology CIBCB, August 12-15 Niagara Falls, ON, Canada, 2015 1 (cit. on pp. 67, 68).

[146] I. T. Ribon,
"Exploiting Semantics from Ontologies to Enhance Accuracy of Similarity Measures",
The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings, 2015 795 (cit. on pp. 67, 68).

[147] P. Resnik, *Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language*,
Journal of Artificial Intelligence Research **11** (1998) 95 (cit. on p. 68).

[148] J. L. Sevilla, V. Segura, A. Podhorski, E. Guruceaga, J. M. Mato, L. A. Martinez-Cruz, F. J. Corrales and A. Rubio,
Correlation Between Gene Expression and GO Semantic Similarity,
IEEE/ACM Trans. Comput. Biol. Bioinformatics **2** (2005) 330, ISSN: 1545-5963 (cit. on p. 68).

[149] F. M. Couto, M. J. Silva and P. M. Coutinho,
Measuring Semantic Similarity Between Gene Ontology Terms,
Data Knowl. Eng. **61** (2007) 137 (cit. on p. 68).

[150] D. Lin, "An Information-Theoretic Definition of Similarity",
Proceedings of the Fifteenth International Conference on Machine Learning, ICML 98, Morgan Kaufmann Publishers Inc., 1998 296, ISBN: 1-55860-556-8 (cit. on p. 68).

[151] J. J. Jiang and D. W. Conrath,
"Semantic similarity based on corpus statistics and lexical taxonomy",
Proceedings of the 10th International Conference on Research in Computational Linguistics, ROCLING 1997, 1997 (cit. on p. 68).

[152] L. Fuenmayor, D. Collarana, S. Lohmann and S. Auer, "FaRBIE: A Faceted Reactive Browsing Interface for Multi RDF Knowledge Graph Exploration",
Proceedings of the Third International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 22. 2017 111 (cit. on p. 71).

[153] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces", *Conference on Human Factors in Computing Systems, CHI 1990, Seattle, WA, USA, April 1-5*, ACM, 1990 249 (cit. on pp. 75, 76).

[154] J. Nielsen, *Usability Engineering*, Academic Press, 1993, ISBN: 9780125184069 (cit. on p. 75).

[155] G. Pirro, "Explaining and suggesting relatedness in knowledge graphs", *ISWC*, Springer, 2015 622 (cit. on p. 76).

[156] E. P. Klement, R. Mesiar and E. Pap, *Triangular norms*, vol. 8, Springer Science & Business Media, 2013 (cit. on p. 77).

[157] K. Gunaratna, K. Thirunarayan, A. P. Sheth and G. Cheng,
"Gleaning Types for Literals in RDF Triples with Application to Entity Summarization",
The Semantic Web. Latest Advances and New Domains - 13th International Conference, ESWC 2016, Heraklion, Crete, Greece, May 29 - June 2, 2016, Proceedings, 2016 85 (cit. on p. 79).

[158] J. Volz, C. Bizer, M. Gaedke and G. Kobilarov,
"Silk - A Link Discovery Framework for the Web of Data",
Proceedings of the Workshop on Linked Data on the Web LDOW Madrid, Spain, April 20, vol. 538, CEUR Workshop Proceedings, CEUR-WS.org, 2009 (cit. on p. 80).

[159] J. Kleinberg and E. Tardos, *Algorithm Design*,
Addison-Wesley Longman Publishing Co., Inc., 2005, ISBN: 0321295358 (cit. on p. 81).

[160] H. W. Kuhn, *The Hungarian method for the assignment problem*,
Naval research logistics quarterly **2** (1955) 83 (cit. on p. 81).

[161] D. Collarana, M. Galkin, C. Lange, S. Scerri, M. Vidal and S. Auer,
"Synthesizing Knowledge Graphs from web sources with the MINTE+ framework",
The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, United States, October 8-12, Proceedings, Part I, 2018 190 (cit. on p. 95).

[162] M. Galkin, D. Collarana, M. Tasnim and M. Vidal,
"Synthesizing a Knowledge Graph of Data Scientist Job Offers with MINTE+",
The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, United States, October 8-12, Proceedings, Part II, 2018 207 (cit. on p. 95).

[163] E. M. Sibarani, S. Scerri, C. Morales, S. Auer and D. Collarana, "Ontology-guided Job Market Demand Analysis: A Cross-Sectional Study for the Data Science field",
Proceedings of the 13th International Conference on Semantic Systems, SEMANTICS 2017, Amsterdam, The Netherlands, September 11-14, 2017 25 (cit. on pp. 95, 104, 105).

[164] P. Ball, *Chemistry: Why synthesize?*, Nature (2015) (cit. on p. 97).

[165] D. Gasevic, D. Djuric and V. Devedzic, *Model Driven Engineering and Ontology Development, 2nd. Ed.* Springer, 2009, ISBN: 978-3-642-00281-6 (cit. on p. 103).

[166] W. Beek, S. Schlobach and F. van Harmelen, "A Contextualised Semantics for owl: sameAs", *Proceedings of the 13th International Conference on The Semantic Web. Latest Advances and New Domains ESWC, Heraklion, Crete, Greece, May 29 - June 2*, 2016 405 (cit. on p. 112).

[167] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute and V. Raghavendra, "Deep Learning for Entity Matching: A Design Space Exploration", *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15*, 2018 19, URL: http://doi.acm.org/10.1145/3183713.3196926 (cit. on p. 113).

[168] X. Bu, J. Rao and C. Z. Xu, "A Reinforcement Learning Approach to Online Web Systems Auto-configuration", *29th IEEE International Conference on Distributed Computing Systems, 22-26 June, Montreal, Quebec, Canada*, 2009 2 (cit. on p. 113).

[169] J. Rao, X. Bu, C. Z. Xu, L. Y. Wang and G. G. Yin, "VCONF: A reinforcement learning approach to virtual machines auto-configuration", *Proceedings of the 6th International Conference on Autonomic Computing, ICAC 2009, June 15-19, Barcelona, Spain*, 2009 137 (cit. on p. 113).

[170] J. C. Barsce, J. A. Palombarini and E. C. Martinez, *Towards Autonomous Reinforcement Learning: Automatic Setting of Hyper-parameters using Bayesian Optimization*, CoRR (2018) (cit. on p. 113).

[171] R. Y. Wang and D. M. Strong, *Beyond Accuracy: What Data Quality Means to Data Consumers*, J. of Management Information Systems **12** (1996) 5 (cit. on p. 113).

[172] I. T. Ribon and M. Vidal, "GARUM: A Semantic Similarity Measure Based on Machine Learning and Entity Characteristics", *Database and Expert Systems Applications - 29th International Conference, DEXA 2018, Regensburg, Germany, September 3-6, Proceedings, Part I*, 2018 169 (cit. on p. 113).

[173] A. Khalili, P. van Andel, P. V. den Besselaar and K. A. de Graaf, "Fostering Serendipitous Knowledge Discovery using an Adaptive Multigraph-based Faceted Browser", *Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017*, 2017 15:1 (cit. on p. 113).

[174] J. Aasman and P. Mirhaji., "Knowledge Graph Solutions In Healthcare For Improved Clinical Outcomes", *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, United States, October 8-12, Proceedings, Part II*, 2018 207 (cit. on p. 114).

APPENDIX A

Complete List of Publications

The following is the complete list of publications peer-reviewed during the development of this Ph.D. thesis.

- *Journal Articles*:
  1. **Diego Collarana**, Mikhail Galkin, Christoph Lange, Maria-Esther Vidal, Sören Auer. *COMET: A COntextualized Molecule-Based intEgration Technique.* In ACM Semantic Web Journal. (To be submitted to the Journal of Web Semantics).

- *Conference Papers*:
  2. **Diego Collarana**, Mikhail Galkin, Christoph Lange, Simon Scerri, Sören Auer, Maria-Esther Vidal. *Synthesizing knowledge graphs from web sources with MINTE^{+}.* In Proceedings of the 17th International Semantic Web Conference (ISWC'18), In-Press;
  3. **Diego Collarana**, Mikhail Galkin, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, Sören Auer. *Semantic Data Integration for Knowledge Graph Construction at Query Time.* In Proceedings of the 11th IEEE International Conference on Semantic Computing (ICSC'17), 109-116;
  4. **Diego Collarana**, Mikhail Galkin, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, Sören Auer. *MINTE: semantically integrating RDF graphs.* In Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics (WIMS'17), 22:1-22:11;
  5. Mikhail Galkin, **Diego Collarana**, Ignacio Traverso-Ribón, Christoph Lange, Maria-Esther Vidal, Sören Auer. *SJoin: A Semantic Join Operator to Integrate Heterogeneous RDF Graphs.* In Proceedings of the 28th International Conference of Database and Expert Systems Applications (DEXA'17), 206-221;
  6. Camilo Morales, **Diego Collarana**, Maria-Esther Vidal, Sören Auer. *MateTee: A Semantic Similarity Metric Based on Translation Embeddings for Knowledge Graphs.* In Proceedings of the 17th International Conference of Web Engineering (ICWE'17), 246-263; **Best Paper Award.**
  7. Mikhail Galkin, Kemele M. Endris, Maribel Acosta, **Diego Collarana**, Maria-Esther Vidal, Sören Auer. *SMJoin: A Multi-way Join Operator for SPARQL Queries.* In Pro-

ceedings of the 13th International Conference on Semantic Systems (SEMANTiCS'17), 104-111;

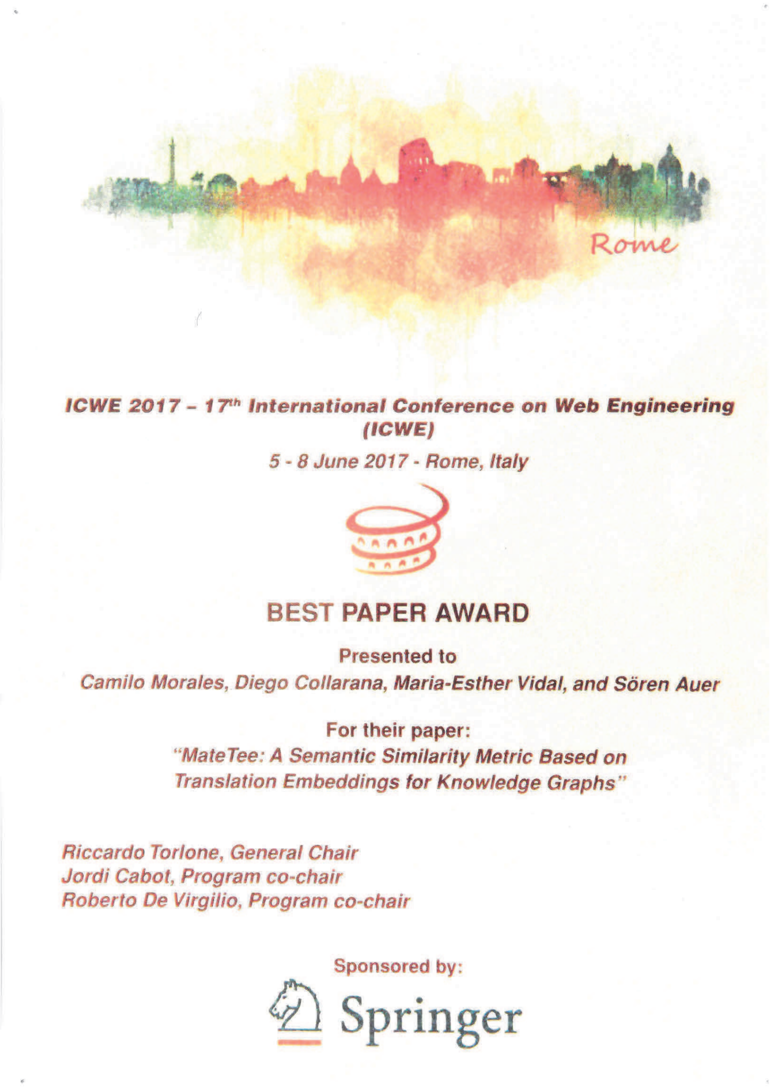
8. Elisa Margareth Sibarani, Simon Scerri, Camilo Morales, Sören Auer, **Diego Collarana**. *Ontology-guided Job Market Demand Analysis: A Cross-Sectional Study for the Data Science field.* In Proceedings of the 13th International Conference on Semantic Systems (SEMANTiCS'17), 25-32;
9. **Diego Collarana**, Mikhail Galkin, Christoph Lange, Irlán Grangel-González, Maria-Esther Vidal, Sören Auer. *FuhSen: A Federated Hybrid Search Engine for Building a Knowledge Graph On-Demand Short Paper.* In Proceedings of the On the Move to Meaningful Internet Systems OTM 2016 Conferences - Confederated International Conferences CoopIS, CTC, and ODBASE (ODBASE'16), 752-761;
10. Irlán Grangel-González, **Diego Collarana**, Lavdim Halilaj, Steffen Lohmann, Christoph Lange, Maria-Esther Vidal, Sören Auer. *Alligator: A Deductive Approach for the Integration of Industry 4.0 Standards.* In Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'17), 272-287;
11. Irlán Grangel-González, Lavdim Halilaj, Gökhan Coskun, Sören Auer, **Diego Collarana**, Michael Hoffmeister. *Towards a Semantic Administrative Shell for Industry 4.0 Components.* In Proceedings of the 10th IEEE International Conference on Semantic Computing (ICSC'16), 230-237;
12. Irlán Grangel-González, Lavdim Halilaj, Sören Auer, Steffen Lohmann, Christoph Lange, **Diego Collarana**. *An RDF-based approach for implementing industry 4.0 components with Administration Shells.* In Proceedings of the 21st IEEE International Conference on Emerging Technologies and Factory Automation (EFTA'16), 1-8;

- *Workshops, Demos, and Doctoral Consortium*:

12. Luis Fuenmayor, **Diego Collarana**, Steffen Lohmann, Sören Auer. *FaRBIE: A Faceted Reactive Browsing Interface for Multi RDF Knowledge Graph Exploration.* In Proceedings of the Proceedings of the Third International Workshop on Visualization and Interaction for Ontologies and Linked Data (VOILA'17), 111-122;
13. **Diego Collarana**. *A Semantic Integration Approach for Building Knowledge Graphs On-Demand.* In Proceedings of the 17th International Conference of Web Engineering (ICWE'17), 575-583;
14. **Diego Collarana**, Christoph Lange, Sören Auer. *FuhSen: A Platform for Federated, RDF-based Hybrid Search.* In Proceedings of the 25th International Conference on World Wide Web (WWW'16), 171-174;

APPENDIX B

Best Paper Award ICWE 2017



Rome

ICWE 2017 – 17th International Conference on Web Engineering (ICWE)

5 - 8 June 2017 - Rome, Italy

BEST PAPER AWARD

Presented to

Camilo Morales, Diego Collarana, Maria-Esther Vidal, and Sören Auer

For their paper:

"MateTee: A Semantic Similarity Metric Based on Translation Embeddings for Knowledge Graphs"

Riccardo Torlone, General Chair
Jordi Cabot, Program co-chair
Roberto De Virgilio, Program co-chair

Sponsored by:

Springer

Figure B.1: Best Paper award at the 17th International Conference on Web Engineering (ICWE), 5 - 8 June 2017, Rome, Italy.

List of Figures

List of Tables