Synthesizing Knowledge Graphs from Web Sources with the MINTE$^+$ Framework

Diego Collarana[1,2(✉)], Mikhail Galkin[1,2,4], Christoph Lange[1,2], Simon Scerri[1,2], Sören Auer[3,6], and Maria-Esther Vidal[2,3,5]

[1] University of Bonn, Bonn, Germany
{collaran,galkin,langec,scerri}@cs.uni-bonn.de
[2] Fraunhofer Institute for Intelligent Analysis and Information Systems, Sankt Augustin, Germany
[3] TIB Leibniz Information Centre for Science and Technology, Hannover, Germany
{soeren.auer,maria.vidal}@tib.eu
[4] ITMO University, Saint Petersburg, Russia
[5] Universidad Simón Bolívar, Caracas, Venezuela
[6] L3S Research Centre, Leibniz University of Hannover, Hannover, Germany

Abstract. 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 the *RDF Molecule-Based Integration Framework 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.

Keywords: Data integration · RDF · Knowledge graphs
RDF molecules

1 Introduction

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 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,

D. Vrandečić et al. (Eds.): ISWC 2018, LNCS 11137, pp. 359–375, 2018.
https://doi.org/10.1007/978-3-030-00668-6_22

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, startups, or SMEs cannot spend comparable resources to collect, integrate, and create value out of such data.

In this paper, we present the use of MINTE$^+$, an RDF Molecule-Based Integration Framework, in three domain-specific applications. MINTE$^+$ is an integration framework that collects and integrates data from heterogenous 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. 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.

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 [8], ontology-based information integration, RDF molecules [4], and semantic annotations [9] to identify relatedness between entities and integrate them into a knowledge graph.

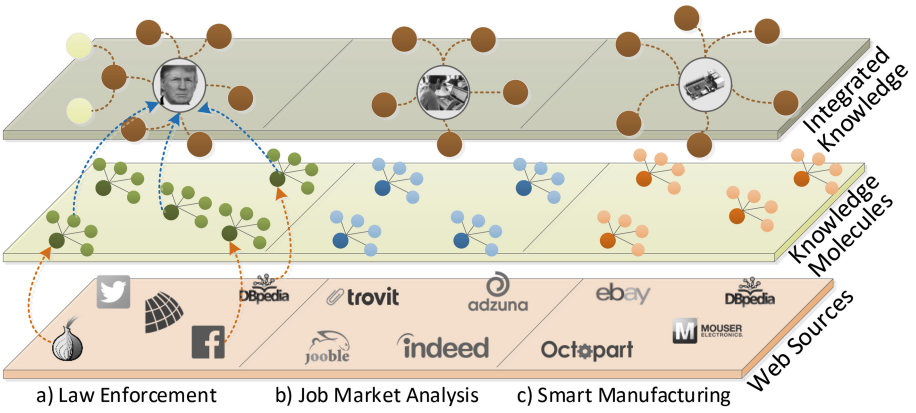


Fig. 1. Domain-specific applications. (a) Law Enforcement agencies need to synthesize knowledge about suspects. (b) For a Job Market analysis, the job offers from different job portals need to be synthesized. (c) A manufacturing company needs synthesized knowledge about providers from web sources.

Motivation: 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 the company's internal knowledge. Figure 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* (top layer of Fig. 1). This knowledge is spread across different web sources such as *social networks*, *job portals*, or Open Knowledge Graphs (bottom layer of Fig. 1). However, the integration 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.

The remainder of the article is structured as follows: Sect. 2 describes MINTE+. Then, the application of MINTE+ in Law Enforcement (Sect. 3), Job Marked Analysis (Sect. 4), and Manufacturing (Sect. 5) is described. Finally, Sect. 6 presents our conclusions and outlines our future work.

2 The Synthesis of RDF Molecules Using MINTE+

Grounded on the semantic data integration techniques proposed by Collarana et al. [3,4], we propose MINTE+, an integration framework able to create, identify, and merge semantically equivalent RDF entities. Thus, a solution to the *problem of semantically integrating* RDF molecules is provided. Figure 2 depicts the main components of the MINTE+ architecture. 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* [2], and the *integration component* integrates RDF molecules into complex RDF molecules in a knowledge graph.

[1] https://www.occrp.org/.

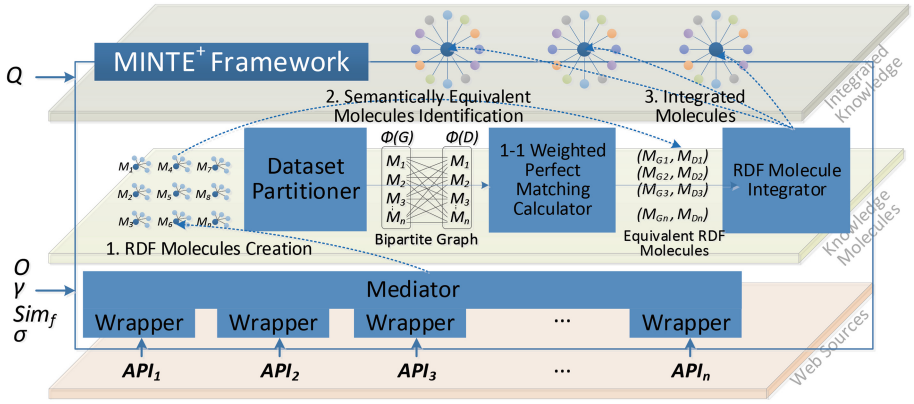


Fig. 2. The MINTE$^+$ Architecture. MINTE$^+$ receives a set of web APIs, a keyword query Q, a similarity function Sim_f, a threshold γ, an ontology O, and a fusion policy σ. The output is a semantically integrated RDF graph.

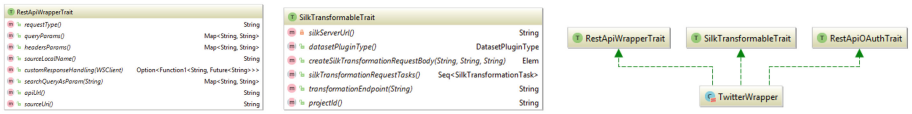
2.1 Creating RDF Molecules

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 takes as much time as the slowest web API. Once a request is 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 3 depicts the interfaces implemented by a wrapper in order to be plugged into the pipeline.

2.2 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 postings are semantically equivalent. A similarity function has to

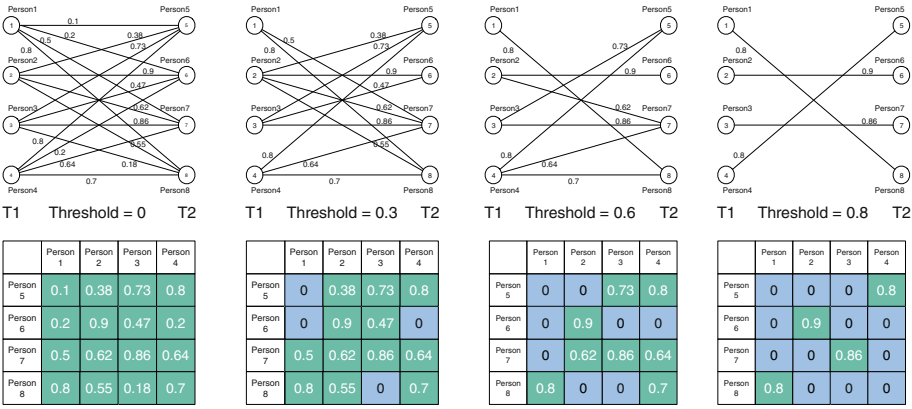
[2] https://developers.google.com/+/web/api/rest/latest/people/search.
[3] http://silkframework.org/.



(a) Web API Interface (b) Silk Interface (c) Twitter Wrapper

Fig. 3. The MINTE+ framework defines three basic interfaces for a Wrapper: WebApiTrait, SilkTransformationTrait, and OAuthTrait.

leverage semantics encoded in the ontology O. For instance, GADES [8] implementation[4] supports this requirement. Additional knowledge about class hierarchy (`rdfs:subClassOf`), equivalence of resources (`owl:sameAs`), and properties (`owl:equivalentProperty`) enables 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.



	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

Fig. 4. Bipartite Graph Pruning. Various thresholds on a semantic similarity function and their impact on creating a bipartite graph between RDF molecules.

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 similarity. We, however, advocate for semantic similarity measures as they achieve better results (as we show in [3]) 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.

Given a bipartite graph $G = (U, V, E)$, the set of vertices U and V are built from two collections of RDF molecules to be integrated (e.g., a wrapper

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

result and an in-memory RDF Graph). Initially, E includes all the edges in the Cartesian product of U and V, and each edge is annotated with the similarity value of the related RDF molecules. In case of a specified threshold, the edges annotated with a similarity value lower than the threshold are removed from E.

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 illustrates how different threshold values affect the number of edges in a bipartite graph. Low threshold values, e.g., 0, result in graphs retaining almost all the edges. Contrarily, when setting a high threshold, e.g., 0.8, graphs are significantly pruned.

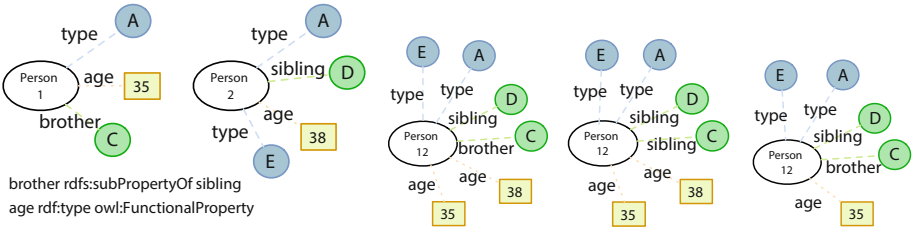
1-1 Weighted Perfect Matching. Having prepared a bipartite graph in the previous step, the *1-1 weighted perfect matching component* identifies the equivalent RDF molecules by matching them with the highest pairwise similarity score; a Hungarian algorithm is used to compute the matching. Figure 4 ($\gamma = 0.8$) 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.

2.3 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., keeping all literals with different language tags or retaining an authoritative one, replacing one predicate with another, or simply merging 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 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. 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 [4]:

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 Fig. 5a the pair $(type, A)$ appears in both molecules. In Fig. 5b, only one pair is retained. The rest of the pairs are added directly.

Subproperty Policy. The policy tracks if a property of one RDF molecule is an `rdfs:subPropertyOf` of a property of another RDF molecule, i.e., $\{r_1, p_1, A\}$, $\{r_2, p_2, B\} + O + subPropertyOf(p_1, p_2) \models \{\sigma_r(r_1, r_2), p_2, \sigma_v(A, B)\}$. 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 Fig. 5c, a property *brother* is generalized to *sibling* preserving the value C according to the subproperty ontology axiom in Fig. 5a.



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

Fig. 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.

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 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 Fig. 5a, *age* has the value 35 in Fig. 5d, 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 `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 thesecriteria.

– 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.

3 A Law Enforcement Application

3.1 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 Fig. 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 Data Web using SPARQL, they may not find the information they need. 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 (in German) "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, in this application no data should be lost, e.g., all aliases or names of a person should be kept.

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

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.

The LiDaKrA[6] project had 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 end users, such as 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, OCCRP, Linked Leaks[7], etc.
U2. Fanaticism and terrorism: searching for advertising, accounts and posts on social networks. Relevant sources are Twitter, Google+, OCCRP, 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.

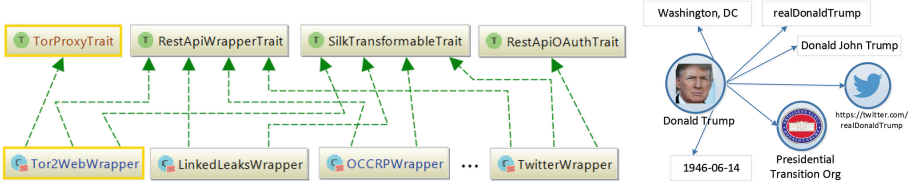
Table 1. MINTE+ Configuration. The Law Enforcement Application

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 [8]	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

[6] https://www.bdk.de/der-bdk/aktuelles/artikel/bdk-beteiligt-sich-im-forschungsprogramm-lidakra.
[7] http://data.ontotext.com/.

3.2 MINTE$^+$ Configuration

To address the challenges of this application and support the use cases, we configured MINTE$^+$ with the parameters shown in Table 1. As keyword Q, the users mainly provide people, organization, or product names, e.g., Donald Trump, Dokka Umarov, ISIS, or Fentanyl. Figure 7a 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 Fig. 3. 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 [8] with a threshold of 0.9. This high value guarantees that only very similar molecules are integrated.



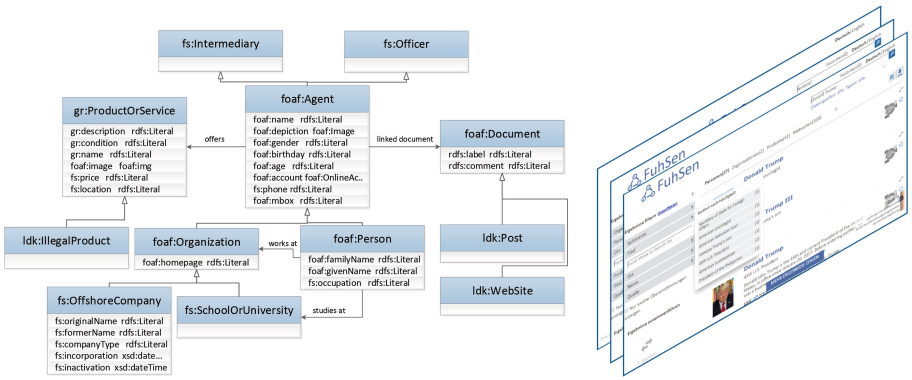
(a) Wrapper Extension for Tor (b) RDF Molecule

Fig. 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.

To address C2, each RDF molecule is annotated with its provenance at creation time using PROV-O[8], Fig. 6b 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.

To close the application cycle, a faceted browsing user interface exposes the integrated RDF graph to users. Figure 7b 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 [1]. In an earlier publication [5], 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.

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



(a) The LiDaKrA ontology (b) User interface

Fig. 7. MINTE+ in LiDaKrA. (a) LiDaKrA UML ontology profile view (cf. [6]) of the main RDF molecule types. (b) The faceted browsing user interface that allows the exploration of the synthesized RDF molecules.

3.3 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 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 furthermore emphasized the importance of 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]

[9] For confidentiality we cannot state their names, nor gather usage data automatically.
[10] https://www.iais.fraunhofer.de/en/business-areas/enterprise-information-integration/federated-search.html.

4 A Job Market Application

4.1 Motivation and Challenges

Declared by Harvard Business Review as the "sexiest job of the 21st century", 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 applying sophisticated big data analysis techniques and producing powerful visualizations. Therefore, it is in the interest of all companies to understand the *job market* and the *skills* demand in 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"[11], is to deliver learning tools that are crucially needed in order to close this problematic skills gap. One of these tools consists of 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 postings need to be annotated with the skills defined in the SARO ontology [9] 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.

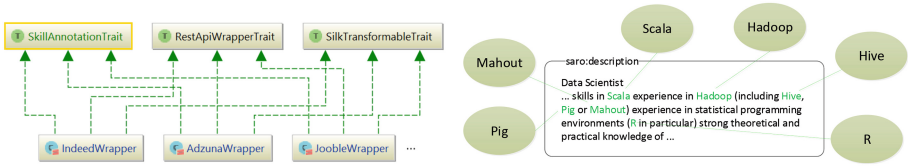
The EDSA dashboard uses the results of the MINTE$^+$ integration framework; it can address the following use cases:

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 top 5 relevant skills for a data scientist at the time of search.

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

Table 2. MINTE+ Configuration. The job market analysis application

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 [9]	The ontology describes data scientist job postings and skills
Web APIs	5	Adzuna, Trovit, Indeed, Jooble, and Xing
Simf	Silk [7]	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



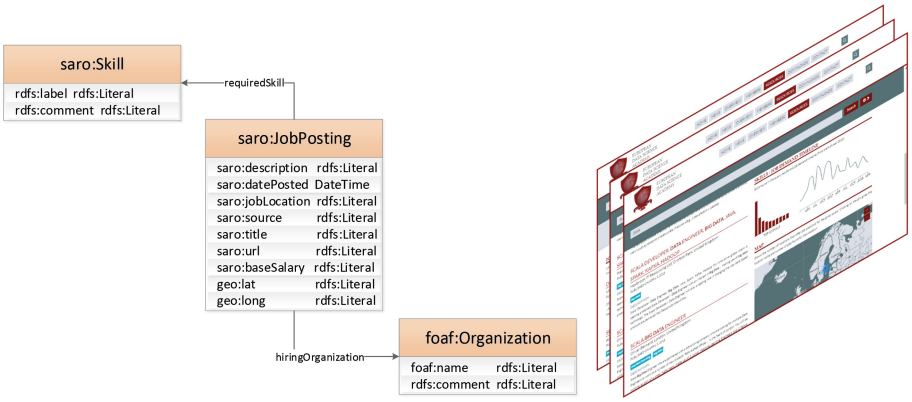
(a) Skill annotation wrapper extension (b) RDF Molecule

Fig. 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.

4.2 MINTE+ Configuration

To address the stated challenges and to support the use cases, we configured MINTE+ with the parameters shown in Table 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 9a depicts the RDF molecule described with the SARO ontology O [9]. To address C1, five wrappers (Adzuna, Trovit, Indeed, Jooble, and Xing) were developed by implementing the interfaces described in Fig. 3. 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 8a shows how the wrappers implement this new interface in addition to the standard ones defined in the framework. Technically, we employ GATE Embedded[12] with a custom REST service[13] to do the annotation using the SARO ontology.

[12] https://gate.ac.uk/family/embedded.html.
[13] https://github.com/EDSA-DataAcquisition/skill-annotation.



(a) SARO ontology (b) User Interface

Fig. 9. MINTE$^+$ in EDSA. (a) The SARO ontology defines the RDF molecules for job market analysis. (b) Screenshot of the EDSA dashboard.

As a similarity function, we resort to Silk [7] 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 postings 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. The EDSA dashboard is then able to use this data to show integrated information about the EU job market.

4.3 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.

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

5 Smart Manufacturing Application

5.1 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: based on the internal metadata of the providers, the company wants to complete their knowledge about them from external sources.

5.2 MINTE+ Configuration

To address the challenges of this application and support the use case, MINTE+ was configured with the parameters shown in Table 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 refers 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`).

5.3 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

[15] For confidentiality reasons we cannot mention the name.
[16] http://schema.org/Organization.
[17] http://schema.org/ContactPage.

Table 3. MINTE$^+$ Configuration. The manufacturing application

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

(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".

6 Conclusions, Lessons Learned, and Future Work

We described the 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 or in production. As global lessons learned we may emphasize that the role of semantic web technology is central to the success of the framework. MINTE$^+$ is able to generate knowledge graphs from remote Web API sources. However, the framework still depends on the quality, consistency, and completeness of the given data. That is, the better the source data, the better is the resulting knowledge graph. We showed the benefits of MINTE$^+$ framework in terms of the configurability and extensibility of its components. The effort to configure, extend and adapt MINTE$^+$ framework 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. Moreover, we are developing machine learning techniques to automatically configure MINTE$^+$ according to the characteristics of the data sources and application domain.

Acknowledgements. Work supported by the European Commission (project SlideWiki, grant no. 688095) and the German Ministry of Education and Research (BMBF) in the context of the projects LiDaKrA ("Linked-Data-basierte Kriminalanalyse", grant no. 13N13627) and InDaSpacePlus (grant no. 01IS17031).

References

1. Arenas, M., Grau, B.C., Kharlamov, E., Marciuska, S., Zheleznyakov, D.: Faceted search over RDF-based knowledge graphs. J. Web Seman. **37**, 55–74 (2016)
2. Ball, P.: Chemistry: why synthesize? Nature **528**(7582), 327 (2015)
3. Collarana, D., Galkin, M., Ribón, I.T., Lange, C., Vidal, M., Auer, S.: Semantic data integration for knowledge graph construction at query time. In: 11th IEEE International Conference on Semantic Computing, ICSC 2017, pp. 109–116 (2017)
4. Collarana, D., Galkin, M., Ribón, I.T., Vidal, M., Lange, C., Auer, S.: MINTE: semantically integrating RDF graphs. In: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, pp. 22:1–22:11 (2017)
5. Collarana, D., Lange, C., Auer, S.: FuhSen: a platform for federated, RDF-based hybrid search. In: Proceedings of the 25th International Conference on World Wide Web, pp. 171–174 (2016)
6. Gasevic, D., Djuric, D., Devedzic, V.: Model Driven Engineering and Ontology Development, 2nd edn. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00282-3
7. Isele, R., Bizer, C.: Active learning of expressive linkage rules using genetic programming. J. Web Seman. **23**, 2–15 (2013)
8. Ribón, I.T., Vidal, M., Kämpgen, B., Sure-Vetter, Y.: GADES: a graph-based semantic similarity measure. In: Proceedings of the 12th International Conference on Semantic Systems, SEMANTICS 2016, pp. 101–104 (2016)
9. Sibarani, E.M., Scerri, S., Morales, C., Auer, S., Collarana, D.: 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 2017, pp. 25–32 (2017)