

Personal Research Agents on the Web of Linked Open Data

Bahar Sateli and René Witte

Semantic Software Lab
Department of Computer Science and Software Engineering
Concordia University, Montréal, QC, Canada

Abstract. We introduce the concept of *Personal Research Agents* as semantics-based entities, capable of helping researchers who have to deal with the overwhelming amount of scientific literature to carry out their daily tasks. We demonstrate how a confluence of state-of-the-art techniques from the Semantic Web and Natural Language Processing domains can realize a proactive agent that can offer *personalized* services to researchers in retrieval and understanding of scientific literature, based on their background knowledge, interests and tasks. The agent’s knowledge base is populated with knowledge automatically extracted from scientific literature of a given domain using text mining techniques and represented in Linked Open Data (LOD) compliant format. Personalization is achieved through automated user profiling, based on a user’s publications. We implemented these ideas in an open source framework and demonstrate its applicability based on a corpus of open access computer science articles.

1 Introduction

“Good morning, Prof. Smith. I found one new publication matching your areas of interest with a contribution that you have not seen before. Would you like to read a summary now?”—Would not it be nice if we all had personal research agents that work around-the-clock, continuously scanning novel research publications and other scholarly communications; agents who know about our daily tasks (reading, reviewing, writing proposals, planning experiments, learning), our interests, even the state of our knowledge in a specific domain, and who can recommend focused information to us? In recent years, the increasing need for an enrichment of scientific literature with semantic metadata has sparked a new series of initiatives in research and development of innovative ways for enhanced scientific dissemination, referred to as *Semantic Publishing* [1]. A recent user survey of scientists, conducted in the context of the *Dr Inventor* EU project [2], revealed that researchers spend almost half of their time locating and reading scientific literature in order to compare their work with other relevant works, highlighting the significant potential for automated support in this area. Semantic publishing aims at making scientific knowledge accessible to both humans and machines, by adding semantic annotations to scholarly content. These annotations are added to research objects, like documents, using special markup with formally defined

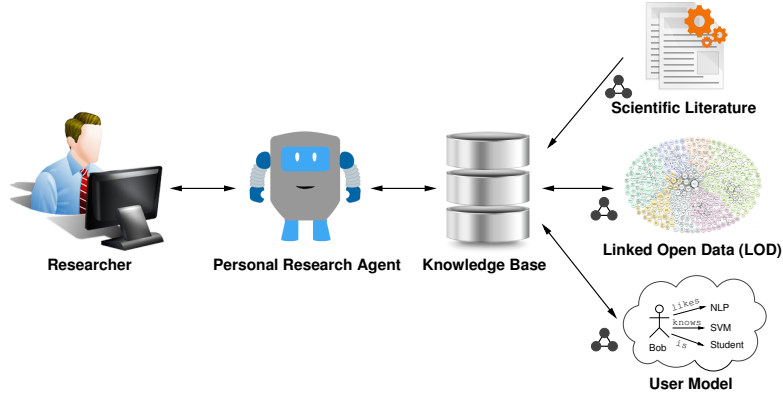


Fig. 1: A high-level overview of our proposed personal research agents

meanings, in order to explicitly mark their structure (e.g., different sections of an article), as well as their *semantics* (e.g., a publication’s contributions, methods, or application domains). However, despite these promises of better knowledge access [3, 4], the manual annotation of existing research literature remains prohibitively expensive for a wide-spread adoption.

In this paper, we investigate how close we are today to the vision of intelligent research agents. Specifically, we build on our previous works in automated text analysis of research articles for their rhetorical structure [5] and the construction of semantic user profiles [6]. Our novel contribution here is the definition of *semantic research agents*, based on Linked Open Data (LOD) principles [7], which are capable of supporting their users through an automatically constructed knowledge base. We analyze the requirements of different user groups for such a personal research agent and formulate services that can satisfy these requirements. The services are then formalized in form of queries against a knowledge base, as shown in Fig. 1. We applied our method on a corpus of 100 open access articles from the *PeerJ Computer Science* journal. All our tools are available as open source software and we published the complete datasets in a GitHub repository at <https://github.com/SemanticSoftwareLab/Supplements-LDK2017>.

2 Literature Review

As mentioned in the introduction, our work is grounded in ideas from the field of semantic publishing, which we review in the following subsection. Work in intelligent agents specifically for the scientific domain is discussed in Section 2.2.

2.1 Semantic Publishing

Semantic publishing, despite its relative infancy, is a fast-paced research domain. Several communities, such as FORCE11,¹ have come together to foster

¹FORCE11, <http://www.force11.org>

research and development towards digital publishing for scholarly communication. Academic events such as conferences, workshops and programming challenges dedicated to text mining of scientific literature are taking place, like the SAVE-SD² (2015–2017) workshops co-located with the International World Wide Web conference, the International Workshop on Mining Scientific Publications³ (2011–2017), the Linked Science Workshops⁴ (2011–2015) and the Semantic Publishing Challenge (2014–2017) in the Extended Semantic Web conference (ESWC).

Moving towards applications for the semantic analysis of scholarly literature, the Semantic Lancet project,⁵ commenced in 2014, aims at making rich data about scholarly publications available as linked open data [8]. Specifically, the Semantic Lancet project goals are (i) to extract structured information from scholarly articles, and (ii) to provide a publicly-available triplestore from the extracted results, upon which a series of value-added services, such as a “*data browser*” or “*abstract finder*” can be implemented.

Semantic Scholar⁶ was developed in 2015 by researchers at the Allen Institute for Artificial Intelligence, as an intelligent search engine to “*cut through the clutter*” when finding computer science literature. Semantic Scholar uses machine learning techniques to find key phrases and citation information from articles and show relevant and “*impactful*” articles by allowing users to filter them using automatically generated facets, like authors or venues.

2.2 Intelligent Scholarly Agents

While intelligent agents and agent-based techniques have a long history in AI, concrete implementations that demonstrate working agents in the domain of science have only started to emerge in recent years.

O’Donoghue et al. [9] introduce a four-level hierarchy of computational processes for sustainable, computationally creative systems that can produce “*new ideas in the form of knowledge or artefacts that represent that knowledge*.” They demonstrate two applications in the Computer Graphics domain that given an input and a goal, can produce *creative* artefacts and processes.

An interesting position paper by Kuhn [10] proposed the concept of “*science bots*” as a model for the future of scientific computation. Kuhn’s bots are autonomous entities that can perform programmed tasks on scholarly data and publish the results. He makes an example of “*a bot [that] could apply text mining to extract relations from the abstracts... and publish the results*”, for example as nanopublications, while “*another one could infer new facts from existing nanopublications by applying specified rules or heuristics*” [10]. Kuhn argues that the bots’ contributions can then be evaluated in a de-centralized way by employing a reputation system, in which humans and other bots can verify the reliability and trustworthiness of the initial bot’s output.

²SAVE-SD workshops, <http://cs.unibo.it/save-sd/>

³WOSP workshops, <https://wosp.core.ac.uk/jcdl2017/>

⁴LISC workshops, <http://linkedscience.org/events/>

⁵Semantic Lancet Project <http://www.semanticlancet.eu>

⁶Semantic Scholar, <https://www.semanticscholar.org/>

Dr Inventor [11] is a European Commission’s Seventh Framework-funded (EU FP7) project⁷ that aims at creating a “*personal research assistant, utilizing machine-empowered search and computation... [to help researchers] by assessing the novelty of research ideas and suggestions of new concepts and workflows.*” The project has received more than 2.6 million Euros and involves multiple university and research institutions from Germany, Spain, Ireland, UK, and Czech Republic. Interestingly, they conducted a survey of researchers [12] on their habits in reading and finding research articles that outlined finding, reading and comparing the rhetorics of different articles with their research goals as the most difficult and time-consuming tasks, which we target to facilitate in this paper.

2.3 Discussion

The retrieval of scientific documents is by now well-supported through numerous Internet search engines, bibliographic databases, and scientific social networks. However, so far there are no tools available that support researchers in some concrete tasks *after* they have retrieved a set of documents (such as, triage, writing a literature review, learning a topic, summarizing a paper). Thus, this is the main goal we want to address with our personal research agents. Albeit very similar in its outlook to create personal research assistants, the focus of the *Dr Inventor* project is to “*promote scientific creativity by utilising web-based research objects*”, specifically for researchers in the Computer Graphics domain. To the best of our knowledge, none of the analysis pipelines developed in that project are available under open source licenses, which is an important contribution of our work.

3 Realizing Personal Research Agents

What distinguishes a personal research agent from other semantic scholarly tools is the ability to construct a flexible, semantically-rich representation of its environment, which can be queried, inferred on and interlinked with external knowledge available on the web of LOD. The three fundamental concepts we need to model in an agent’s knowledge base are *(i)* scholarly documents, *(ii)* users with various backgrounds and information needs, and *(iii)* tasks that the agent is capable of conducting. In this section, we elaborate on how we can automatically construct and exploit such a knowledge base, such that the agent can provide various scholarly services.

3.1 Semantic Modeling of Scientific Literature

Scholarly documents are published in various formats (e.g., PDF, HTML) and typesettings (e.g., ACM, LNCS) optimized for human-reading. In contrast, the agent – as a machine – maintains a different representation of documents in its knowledge base. In our approach, we use the plain-text of documents to extract specific entities that can help the agent to understand their meaning.

⁷Dr Inventor Project, <http://drinventor.eu>

Table 1: Terms from linked open vocabularies for semantic agent modeling

LOV Term	Modeled Concept
bibo:Document	A class that represents scholarly documents.
doco:Sentence	A class that represents sentences in a document.
sro:RhetoricalElement	A class used to classify sentences containing a rhetorical entity, e.g., a <i>contribution</i> or <i>claim</i> .
pubo:LinkNamedEntity	A class to represent document topics, which are linked to their corresponding LOD resource.
cnt:chars	A property to store the verbatim content of entities as they appeared in the document.
pubo:hasAnnotation	A property to relate annotations (e.g., named entities) to documents.
pubo:containsNE	A property to relate rhetorical and named entities in a document.
oa:start & oa:end	A property to show the start and end offsets of entities in a document's text.
um:User	A class to represent scholar users.
c:Competency	A class to represent authors' competence topics (LOD resources) in their publications.
c:CompetenceRecord	A class to record the metadata of authors' competences (e.g., provenance, LOD resource).
um:hasCompetencyRecord	A property to assign a competence record to the user.
c:competenceFor	A property to represent the relation between a competence record and the competence topic.

um: <http://intelleo.eu/ontologies/user-model/ns/> c: <http://intelleo.eu/ontologies/competences/ns/>
cnt: <http://www.w3.org/2011/content#> sro: <http://salt.semanticsauthoring.org/ontologies/sro#>
pubo: <http://lod.semanticssoftware.info/pubo/pubo#> rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
oa: <http://www.w3.org/ns/oa/> rdfs: <http://www.w3.org/2000/01/rdf-schema#>
bibo: <http://purl.org/ontology/bibo/> doco: <http://purl.org/ontology/bibo/>

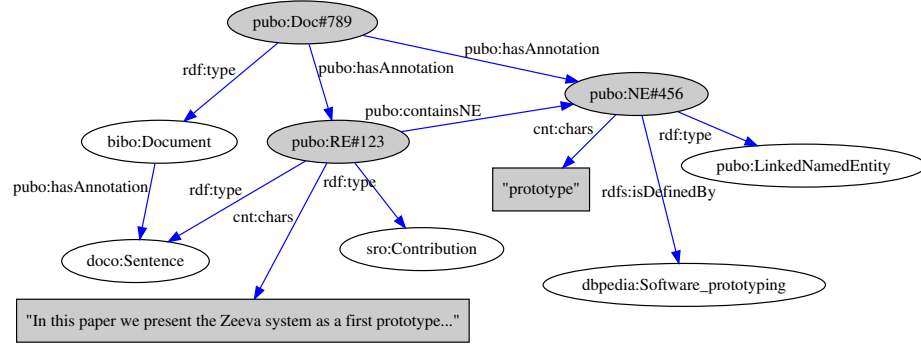


Fig. 2: Agent's model of relations between documents and topics using RDF

Design. The agent's knowledge base is populated with information extracted from input documents using our approach described in [5]. Our workflow transforms scholarly articles to semantic triples using a text mining pipeline. The generated triples contain various structural and semantic elements. Structural entities encompass mostly the bibliographical metadata, such as title and authorship, while the semantic entities are concerned with the *meaning* of the document. The meaning of a document is modeled as a set of selected sentences highlighting its authors contributions (rhetorics). As we demonstrated previously in [5], the collective set of topics mentioned within these rhetorical zones can be used to model the meaning of an article. Fig. 2 shows the vocabularies used to model documents in the agent's knowledge base. We use a selected set of vocabularies from BIBO, DOCO, SRO and PUBO ontologies, as explained in Table 1. The shaded resources in Fig. 2 are example instances of the agent's document schema.

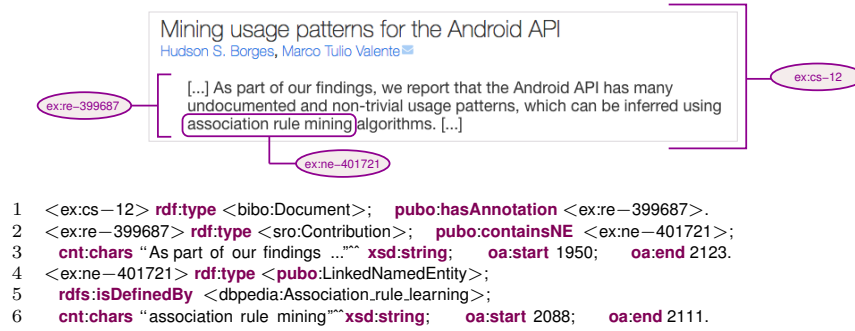


Fig. 3: Example triples (bottom) generated from a document’s sentence (top)

Implementation. Our agent’s knowledge base is populated with document entities using our text mining pipeline described in [5]. Developed based on the GATE framework [13], it accepts (English) articles in PDF or XML formats and stores the generated triples in a TDB-based⁸ triple store. It uses GATE’s ANNIE plugin [13] to pre-process the documents. To extract the rhetorical entities (REs), documents are analyzed by our Rhetector⁹ plugin that can classify each sentence into one of **Claims**, **Contributions** or *neither* categories (with 0.73 F-measure [5]). Document’s topics (in form of linked named entities) are spotted using the LODtagger¹⁰ plugin that acts as a wrapper for the DBpedia Spotlight [14] named entity recognition service. Every mention of a topic (named entity) is tagged with a semantic type and linked to its corresponding LOD resource using a uniform resource identifier (URI). In contrast to REs that span over one or more sentences, NEs are nouns and noun phrases. Finally, all extracted (RE and NE) entities are stored in the knowledge base, using our LODeXporter¹¹ component.

Example. To demonstrate how a document is represented in the agent’s knowledge base, we chose a random document from the dataset in our supplementary materials and analyzed it with our text mining pipeline [5]. Fig. 3 shows an excerpt of the resulting triples in Turtle syntax. The namespaces shown in the listing can be resolved using Table 1. Line 1 of the listing represents the document (doi:10.7717/peerj-cs.12) in the knowledge base, along with a rhetorical entity (sentence) that was found in its text. Lines 2-3 and 4-6, respectively, model the rhetorical entity classified as a **sro:Contribution** and one of the named entities (topics) mentioned in the sentence, i.e., **<dbpedia:Association_rule_learning>**. The corresponding sentence from the original document is shown in Fig. 3 (top).

⁸Apache Jena TDB, <https://jena.apache.org/documentation/tdb/>

⁹Rhetector, <http://www.semanticsoftware.info/rhetector>

¹⁰LODtagger, <http://www.semanticsoftware.info/lodtagger>

¹¹LODeXporter, <http://www.semanticsoftware.info/lodexporter>

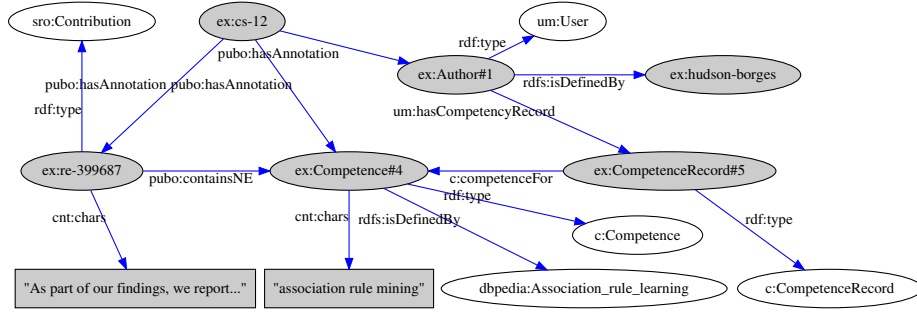


Fig. 4: An RDF graph representing a user profile in the agent's model

3.2 Semantic Modeling of Scholarly Users

While most modern information retrieval tools can help users to semantically expand or restrict a set of result documents, the personalized aspect of research agents provides for value-added features, like showing a ranked list of documents, based on how *relevant* or *novel* they are for a user. It can also be used to help a user, like a student, to understand previously unseen topics when reading them. This, in turn, requires the agent to have a detailed model of a user's context, in particular for background knowledge (what the user already knows) and the task at hand (what information is needed right now). In our agent's design, we store these information in so-called *scholarly user profiles*.

Design. Our semantic representation of user profiles is inspired by the IntelLEO framework¹² for modeling learning contexts. The idea here is to record the background knowledge of a scholar as a set of *competences*. User's competences are defined using a *competence record* that stores the metadata of how and where it was inferred from (e.g., a sentence in a document), as well as a *competence topic* that specifies the relevant entity (e.g., 'LOD'). Constructing user profiles requires collecting user information over an extended period of time and suffers from the *cold-start* problem, where not enough information about the user is available at the beginning to make meaningful recommendations. Since asking users to populate their user profiles with potentially thousands of topics that they know about or are interested in is impractical, we bootstrap users' profiles using their publication history [6]. For each user, we process his publicly available publications to automatically extract his relevant competences. Note that such a model follows a closed-world assumption, i.e., if a topic is not available in a user's profile, the agent will consider it as novel or unknown to the user. Fig. 4 shows the agent's user profile schema with shaded nodes as instances.

Implementation. The assumption in our user modeling is that, if a user has authored a publication on a topic, the user is most likely competent in that topic

¹²IntelLEO framework, <https://www.intelleo.eu/ontologies/learning-context/spec/>

```

1 <ex:Author#1> rdf:type <um:User> ; rdfs:isDefinedBy <ex:hudson—borges> ;
2 um:hasCompetencyRecord <ex:CompetenceRecord#5> .
3 <ex:CompetenceRecord#5> rdf:type <c:CompetenceRecord> ; c:competenceFor <ex:Competence#4> .
4 <ex:Competence#4> rdf:type <c:Competence> ;
5 rdfs:isDefinedBy <http://dbpedia.org/resource/Association_rule_learning> ;
6 cnt:chars "association rule mining" xsd:string ; oa:end 2111 ; oa:start 2088 .

```

Fig. 5: Example user competence record generated from a document sentence

to various degrees. Therefore, using our approach in [6], for each user we analyze their publications with our text mining pipeline. We customized the text mining pipeline, such that the topics (named entities) within the rhetorical zones of a user’s publications are stored as his competences, along with their term frequency using a selected set of terms from the IntelLEO ontology.

Example. We generated a user profile for the first author shown in Fig. 3 as an example. An excerpt of the generated profile is listed in Fig. 5. Line 1 and 3, respectively, model the author and his competence record. Lines 4-6 describe his competence in ‘*association rule learning*’, because it was mentioned in his publication. An advantage of modeling topics as LOD named entities is that (i) user profile competence topics can be integrated with the agent’s modeling of documents, and (ii) if authors use different surface forms of the same topic (e.g., ARL instead of association rule mining), they can be resolved to the same semantics. In this example, resource <ex:Competence#4> represent the same topic as <ex:ne-401721> in Fig. 3, since they are both defined by the same DBpedia resource (http://dbpedia.org/resource/Association_rule_learning).

3.3 Semantic Modeling of Agent’s Tasks

We now introduce our new model for a semantic description of the workflow between a scholar and his personal research agent. While document models and user profiles in the knowledge base are populated as the user interacts with his agent, the metadata and services of the agent are mostly modeled up-front and may be extended throughout the agent’s lifecycle. A formal semantic description of tasks facilitates consistent implementation of the agent’s services and allows for composing new services by combining various tasks that an agent can perform.

Our Personal Research Agent Vocabulary (PRAV)¹³ is an adaptation of the Lifecycle Schema,¹⁴ which was originally designed to model the lifecycle of any resource throughout a transition. Following the best practices of LOD design [7], we tried to re-use existing linked open vocabularies to the extent possible.

Design. An agent’s work unit is a *Task* assigned to it by a user. Tasks are aggregated into *Task Groups* and can be composed in an ordered sequence. While

¹³Personal Research Agent Vocabulary, <http://lod.semanticsoftware.info/prav/prav#>

¹⁴Lifecycle Schema, <http://vocab.org/lifecycle/schema#>

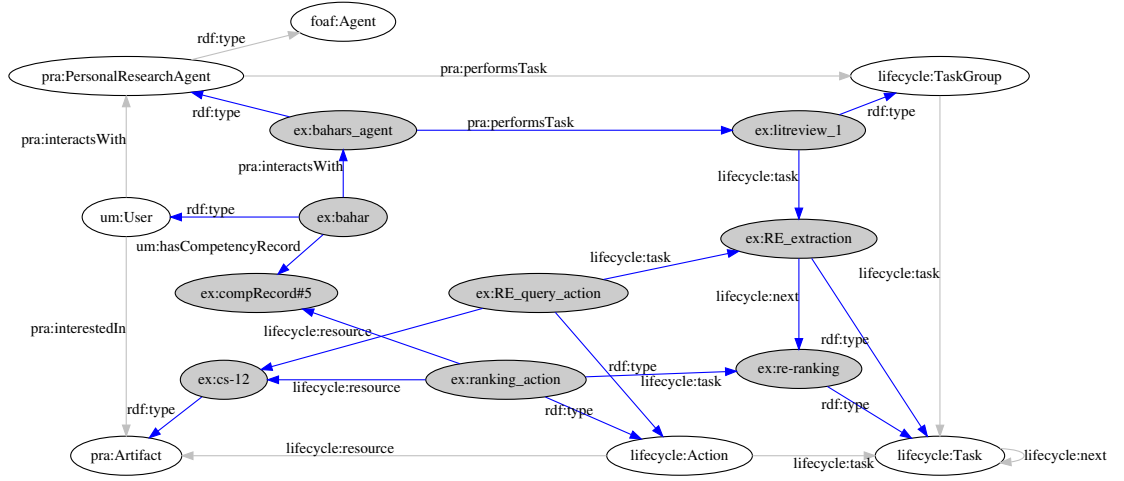


Fig. 6: Example literature review task modeling using the agent's task model

tasks are essentially conceptual entities with properties, such as a description or status, the underlying computations are instances of the *Action* class. Whereas tasks are designed by the agent developers for a specific goal, actions are generic operations, like querying the knowledge base or crawling a repository. In the process, actions can consume, produce or modify *artifacts*. In this paper, we restrict our agent's design to analyze scholarly literature (e.g., journal articles or conference proceedings) as artifacts. Fig. 6 shows our agent's task schema, as well as example instances. For example, a literature review task group shown in the model is divided between two consequent tasks: (i) finding all rhetorical entities from documents mentioning a topic, and (ii) given a user profile, re-ranking the result documents based on how interesting they are for the user. As seen in the agent's schema, certain actions like `<ex:ranking_action>` need access to the knowledge available both within documents and a user's competence records.

Implementation. As a concrete implementation of our agent's tasks, we defined three semantic scholarly services and formulated them as a set of queries. The queries are hand-crafted and implemented using SPARQL syntax. There are two types of queries in our design: (i) queries looking for *concepts*, like finding all things of type `<bibo:Document>` in the knowledge base, and (ii) queries that can be parameterized, such as finding all **Contribution** sentences mentioning '*linked open data*'. Wherever the required knowledge does not readily exist in the agent's knowledge base, but may be available on the web of LOD, we incorporated federated queries to integrate additional information from external resources. In particular, we query the DBpedia ontology¹⁵ through its SPARQL endpoint.

¹⁵DBpedia Ontology, <http://wiki.dbpedia.org/services-resources/ontology>

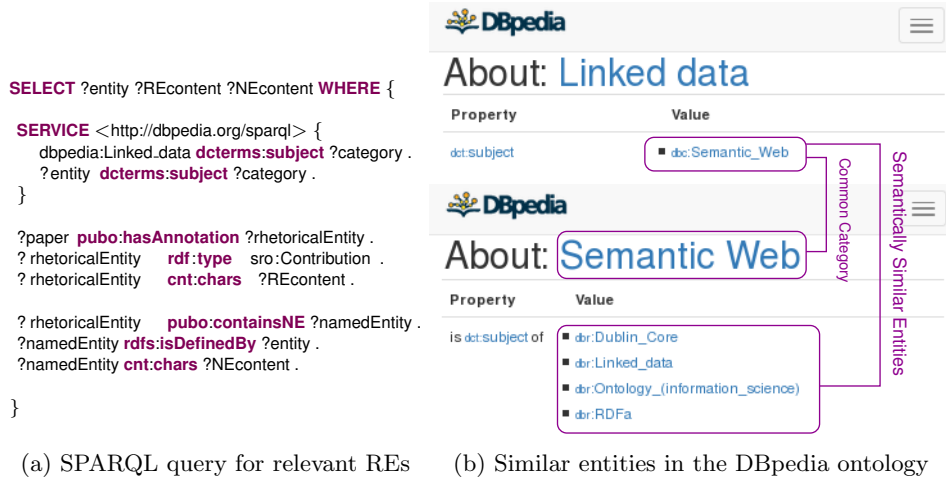


Fig. 7: Agent’s query to find semantically relevant topics for query expansion

Example. As part of our contribution, we define a number of research-related semantic services and demonstrate how our personal research agent can offer them to a user. The services described below by no means form an exhaustive list – a multitude of other services can be provided to users by exploiting the agent’s knowledge base, simply by virtue of defining additional queries.

S1: Summarizing Relevant Articles. Our agent can help researchers in finding and reading scientific literature, by showing only parts that are interesting for a task, like a literature review. One way of obtaining such a summary is by listing all **Contributions** or **Claim** sentences from the document mentioning specific topics. This way, users can examine the rhetorical zones of a document prior to deciding whether they need to read the full-text paper. Fig. 7 shows the agent’s query that can retrieve such results from its knowledge base. Note that since the topics in the knowledge base are essentially LOD resources, by traversing the LOD cloud, our agent can expand the user-provided topics and bring in topics that are semantically related entities from external sources, thus showing a broadened set of relevant documents that may not directly mention the user topics but contain similar entities. In this case, we defined semantic similarity between two resources as being under the same DBpedia category, as shown in Fig. 7b.

S2: Curating a Personalized Reading List. To demonstrate the personalization aspect of our research agent, we formulated a query that can integrate a user profile in order to re-rank a set of documents retrieved from a service like *Summarization* (S1). The idea here is that the number of matching topics (i.e., named entities in a document and competence topics in a user profile) can be used as a means to rank a document in terms of its *interestingness* for the user. Fig. 8 shows how the agent calculates the total number of matching topics

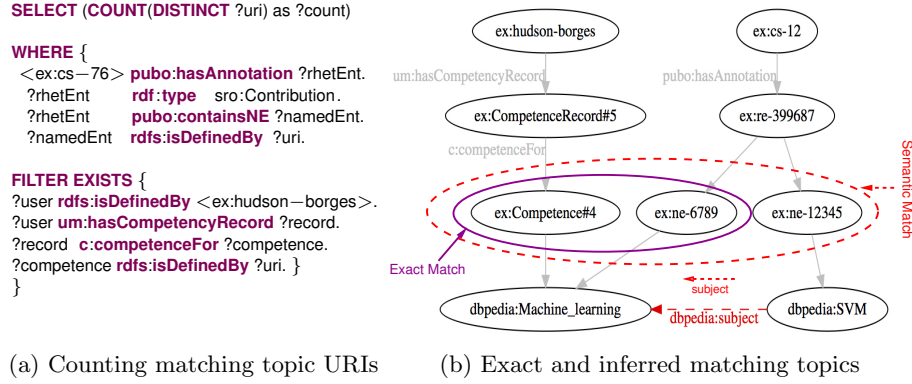


Fig. 8: Counting the number of common topics between a paper and user profile

in the contribution sentences of document `<ex:cs-78>` and user `<ex:HUDSON-BORGES>`' competence records from his profile, by examining their matching LOD URIs. By incorporating the semantic expansion of topics using the query shown in Fig. 7a, the agent can also find topics that may not be mentioned in the user competence record, but fall under the same category in the DBpedia ontology.

S3: Filling the Knowledge Gap of Learners. Another task to demonstrate the advantage of using personal research agents is to offer contextual help to users, for example, when reading a document. The goal of this service is to provide the user with a brief explanation of terms that he is not familiar with as the user is reading the article. Here the agent interprets *new* topics as named entities that do not exist in his knowledge model of the user's competences. The SPARQL query shown in Fig. 9 allows the agent to detect these previously unseen topics in a document and retrieve a brief description from the DBpedia knowledge base.

4 Experiments

As a concrete application of our personal research agent, we performed a number of experiments using a set of open access articles from a computer science journal.

4.1 Dataset

We downloaded a subset of articles from the computer science edition of *PeerJ Computer Science* journal. The dataset contains 100 articles with an average length of 23.25 pages. Since additional metadata is available in the PeerJ XML article format, for each document we retrieved its corresponding XML file and processed them with the text mining pipelines for document analysis and user profile construction, described in Sections 3.1 and 3.2, respectively.



(a) Federated query for external knowledge (b) Available information from DBpedia

Fig. 9: Agent’s query for topics in the document that are unknown to the user

Table 2: Quantitative analysis of the agent’s knowledge base after population

Entity Type	#Total in KB	#Average per document	
		Mean	Standard Deviation (σ)
RDF Triples	1,281,971	3,131.93	1,394.92
Sentence	135,980	1,359.80	779.65
Contribution	1,241	12.41	16.81
Claim	421	4.21	4.07
Linked Named Entity	144,611	1,446.11	627.10
User	395	3.95	3.70
Competence Record	30,545	305.45	627.12

4.2 Knowledge Base

We populated a TDB-based knowledge base with the text mining pipelines’ output, resulting in a total of 1,281,971 RDF triples. The complete processing time for the analysis of the documents and their authors’ competences took 16.05 minutes on a MacBook Pro 2.5 GHz Intel Core i7 with 16 GB memory, with DBpedia Spotlight taking up to 55% of the processing time. For each author of the dataset documents, we populated a user profile with competence records extracted from their corresponding documents’ rhetorical zones. User profile triples are merged with the document models through common named entity URIs. Table 2 shows a quantitative analysis of the agent’s knowledge base.

4.3 Queries

In this section, we revisit the scholarly services that our agent can offer its users and show a number of actual outputs from our experiment’s dataset.

Documents matching <code>dbpedia:Linked_data</code> and <code>dbpedia:Academic_publishing</code> :
Document ID: http://example.com/papers/cs-78
Title: Decentralized provenance-aware publishing with nanopublications (Tobias Kuhn et al.)
Matched Entities:
<ul style="list-style-type: none"> • <code>dbpedia:Linked_data</code> <i>Inferred matches:</i> <code>dbpedia:Ontology_(information_science)</code> , <code>dbpedia:Semantic_Web</code> , <code>dbpedia:Controlled_vocabulary</code> , ...
<ul style="list-style-type: none"> • <code>dbpedia:Academic_publishing</code> <i>Inferred matches:</i> <code>dbpedia:Proceedings</code> , <code>dbpedia:Literature_review</code> , <code>dbpedia:Open_access</code> , ...
Selected Snippets to Read:
"We show how this approach allows researchers to publish, retrieve, verify, and recombine datasets of nanopublications in a reliable and trustworthy manner, and we argue that this architecture could [...] serve the <i>Semantic Web</i> in general."

Fig. 10: The agent's output assisting a researcher in a literature review task

S1: Summarizing articles on ‘LOD’ and ‘Academic Publishing’. Let us imagine a user asks his agent to generate a summary of relevant articles on “*the integration of linked open data vocabularies in academic publishing*” from the PeerJ dataset. The agent first processes the user input and finds two entities, namely, `<dbpedia:Linked_data>` and `<dbpedia:Academic_publishing>`. It then queries its knowledge base to find documents that mention any or both of these entities within their rhetorical zones using the query in Fig. 7. Additionally, the agent service results will not just be a list of matching documents, rather it will determine sentences of the documents the user needs to read. Fig. 10 shows an example output produced by the agent for this query.

S2: A personalized reading list from S1 output. Using the agent's service in the previous section, our user receives 34 articles in the PeerJ dataset that matches either `<dbpedia:Linked_Data>`, `<dbpedia:Academic_publishing>`, or both, as well as any semantically relevant entities to his query terms. Next, he asks the agent to personalize his results according to their *interestingness*. Here, the agent assumes an article is interesting for a user, if there are matching named entities both in an article and in the users' profile. The agent then sorts the results from the first service based on the total number of common topics between each article and the users profile competences, in descending order, as retrieved by the SPARQL query shown in Fig. 8. We show an example output in Fig. 11.

S3: Suggesting background knowledge during reading tasks. While our user is reading an article from his personalized list, whenever he encounters a new topic, the agent can retrieve a brief description of the topic from available ontologies, point him to its reference Wikipedia page, or better, retrieve passages from other documents in the knowledge base on how, e.g., a methodology is used in practice. For example, the agent can identify topics that the user does not know about (i.e., absent from his profile) and suggest background knowledge by executing the query shown in Fig. 9. We show an example output in Fig. 12.

Documents matching <code>dbpedia:Linked_data</code> and <code>dbpedia:Academic_publishing</code> :
1. Document ID: http://example.com/papers/cs-78
Title: A technology prototype system for rating therapist empathy from audio recordings in addiction [...] (Tobias Kuhn et al.)
Matching interests in your profile:
<code>dbpedia:Semantic_Web</code> <code>dbpedia:Data</code> , <code>dbpedia:Publishing</code> , <code>dbpedia:First-order_logic</code> , <code>dbpedia:Client-server_model</code> , ...
2. Document ID: http://example.com/papers/cs-64
Title: Enriching scientific publications with interactive 3D PDF: an integrated toolbox for creating ready [...] (Axel Newe)
Matching interests in your profile:
<code>dbpedia:Academic_publishing</code> , <code>dbpedia:Document</code> , <code>dbpedia:Software</code> , <code>dbpedia:Portable_Document_Format</code> , ...

Fig. 11: A personalized reading list with matching interests as explanation

Document ID: http://example.com/papers/cs-78
Title: A technology prototype system for rating therapist empathy from audio recordings in addiction [...] (Tobias Kuhn et al.)
Selected Snippets to Read:
<ul style="list-style-type: none"> • “In this article, we propose to design scientific data publishing as a web-based bottom-up process, without top-down [...]” • “We argue that the centralized nature of existing data repositories is inconsistent with the <u>decentralized</u> manner in [...]”
Topics New to You:
<ul style="list-style-type: none"> • <i>Top-down and bottom-up design</i>: Top-down and bottom-up are both strategies of information processing and knowledge ordering, used in a variety of fields including [...]. Source: https://en.wikipedia.org/wiki/Top-down_and_bottom-up_design

Fig. 12: Personal agent assisting researchers in understanding unknown topics

5 Conclusion and Future Work

The amount of knowledge available in digital libraries is still increasing at a rapid pace. While finding and accessing scientific documents has become easier in recent years, thanks to various search engines and bibliographic services, the most labor-intensive tasks of reading and evaluating these search results still remains largely unsupported. In this paper, we proposed *personal research agents* that support their human users in knowledge-intensive, research-related activities.

While this idea been envisioned for more than a decade now, we examined the concrete support that can be realized with available technologies today: In our approach, we first transform scientific articles from natural language documents living in isolation into queryable knowledge bases with explicit semantics. Orthogonal to this, we formally model scientific users, their background knowledge, projects and tasks that they carry out in their day-to-day research activities. The synthesis of these individual knowledge bases then serves as the ‘brain’ of our personal research agents. With this approach, we can formulate a number of complex tasks performed daily by researchers, students, editors, or reviewers, in form of knowledge base queries. The presented ideas are fully implemented and our open source pipelines can be immediately deployed by anyone who wants to start running their own personal research agent. In future work, we will perform

a number of user studies to examine how end users interact with their agents, and evaluate how they improve manually performed tasks in a scientific workflow.

References

1. Peroni, S.: *Semantic Web Technologies and Legal Scholarly Publishing*. Springer (2014)
2. Inventor, D.: *Promoting Scientific Creativity by Utilising Web-based Research Objects*. Technical report (February 2014) Program no. FP- ICT- 2013.8.1.
3. Berners-Lee, T., Hendler, J.: Publishing on the semantic web. *Nature* **410**(6832) (2001) 1023–1024
4. Shotton, D.: Semantic publishing: the coming revolution in scientific journal publishing. *Learned Publishing* **22**(2) (2009) 85–94
5. Sateli, B., Witte, R.: Semantic representation of scientific literature: bringing claims, contributions and named entities onto the Linked Open Data cloud. *PeerJ Computer Science* **1**(e37) (12/2015 2015)
6. Sateli, B., Löffler, F., König-Ries, B., Witte, R.: Semantic User Profiles: Learning Scholars’ Competences by Analyzing their Publications. In Gonzalez-Beltran, A., Osborne, F., Peroni, S., eds.: *Semantics, Analytics, Visualisation: Enhancing Scholarly Data (SAVE-SD 2016)*. Volume 9792., Springer (04/2016 2016)
7. Heath, T., Bizer, C.: *Linked Data: Evolving the Web into a Global Data Space*. Synthesis lectures on the semantic web: theory and technology. Morgan & Claypool Publishers (2011)
8. Bagnacani, A., Ciancarini, P., Di Iorio, A., Nuzzolese, A., Peroni, S., Vitali, F.: The Semantic Lancet Project: A Linked Open Dataset for Scholarly Publishing. In Lambrix, P., Hyvönen, E., Blomqvist, E., Presutti, V., Qi, G., Sattler, U., Ding, Y., Ghidini, C., eds.: *Knowledge Engineering and Knowledge Management*. Volume 8982 of *Lecture Notes in Computer Science*. Springer (2015) 101–105
9. O’Donoghue, D.P., Power, J., O’Briain, S., Dong, F., Mooney, A., Hurley, D., Abgaz, Y., Markham, C.: Can a Computationally Creative System Create Itself? Creative Artefacts and Creative Processes. In: *5th International Conference on Computational Creativity*, Jožef Stefan Institute (2014)
10. Kuhn, T.: Science Bots: a Model for the Future of Scientific Computation? In: *Proceedings of the 24th International Conference on World Wide Web Companion*. (2015) 1061–1062
11. Ronzano, F., Saggion, H.: Dr. Inventor Framework: Extracting Structured Information from Scientific Publications. In: *Discovery Science*, Springer (2015) 209–220
12. Chaudhry, E., Yang, X., You, L.: *Promoting Scientific Creativity by Utilising Web-based Research Objects: Deliverable No. 2.1 User requirement and use case report* (2014)
13. Cunningham, H., et al.: *Text Processing with GATE (Version 6)*. (2011)
14. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. In: *Proc. of the 7th International Conf. on Semantic Systems*, ACM (2011) 1–8