



A Semantic Web Navigation Tool for Exploring the Henri Poincaré Correspondence Corpus

Nicolas Lasolle, Olivier Bruneau, Jean Lieber, Philippe Nabonnand, Laurent
Rollet

► To cite this version:

Nicolas Lasolle, Olivier Bruneau, Jean Lieber, Philippe Nabonnand, Laurent Rollet. A Semantic Web Navigation Tool for Exploring the Henri Poincaré Correspondence Corpus. Proceedings of the International Joint Workshop on Semantic Web and Ontology Design for Cultural Heritage, Antonis Bikakis, Roberta Ferrario, Stéphane Jean, Béatrice Markhoff, Alessandro Mosca, Marianna and Nicolosi Asmundo, Sep 2021, Bolzano, Italy. hal-03406713

HAL Id: hal-03406713

<https://hal.univ-lorraine.fr/hal-03406713>

Submitted on 28 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Semantic Web Navigation Tool for Exploring the Henri Poincaré Correspondence Corpus

Nicolas Lasolle^{1,2}[0000–0002–1253–649X], Olivier Bruneau¹, Jean Lieber²,
Philippe Nabonnand¹, and Laurent Rollet¹

¹ Université de Lorraine, CNRS, Université de Strasbourg, AHP-PreST, F-54000
Nancy, France

{firstName.lastName}@univ-lorraine.fr

² Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{firstName.lastName}@loria.fr

Abstract. The Henri Poincaré correspondence corpus is composed of around 2200 letters which constitute a rich source of information for historians of science. The following Semantic Web technologies have been chosen to represent and to exploit corpus data: the RDF model, the RDFS knowledge representation language and the SPARQL query language. Recently, the need of an exploration tool for this historical corpus has emerged. As a result, a navigation tool has been created to explore the correspondence by exploiting the similarities between the documents of the corpus. This system relies on the use of a SPARQL query transformation rule mechanism which can help to highlight unexpected relations between resources. This article presents the context of this work and illustrates the developed system through a concrete use case for the Henri Poincaré correspondence corpus. The system architecture and some reusability elements are presented, as the tool can be connected to other corpora, as long as underlying data is exposed through a public SPARQL endpoint.

Keywords: Digital humanities · Semantic Web · Historical corpora · SPARQL query transformation · Knowledge base exploration

1 Introduction

Jules Henri Poincaré is an important figure in the history of science. Born in Nancy, France, in 1854, he died in Paris in 1912. He is famous for several significant contributions in multiple areas of mathematics and physics [14]. Henri Poincaré also had an interest in philosophy, especially in philosophy of science, as shown by several of his works presented in books such as *La Science et l'Hypothèse* [15]. During his career, Henri Poincaré held many roles within French administrations and learned societies such as the French Academy of Sciences or the *Bureau des longitudes*. He was also involved in international societies—mainly as a corresponding member—such as The Royal Society or the Berlin Academy of Science.

Among the research works dedicated to this man of science, there is a keen interest in the corpus of his correspondence which is composed of around 2200 letters. This corpus gathers private, academic and scientific exchanges which are of interest to historians of science. The study of this corpus is a long-term project which has led to the publication of several thematic volumes. For instance, the last volume is dedicated to his youth [17], when he was a student at the *École polytechnique* and then at the *École des mines* (1873-1878). Another example is the volume which gathers his regular exchanges with the Swedish mathematician Gösta Mittag-Leffler [13]. An important aspect of this project is related to the online publishing and exploitation of this corpus.

The Henri Poincaré website³ is managed thanks to the Omeka S content management system [3], a tool that enables the Web publishing and the sharing of cultural collections from institutions (museums, archive places, etc.). The letters of the corpus are available on this platform and come with a scan of the original document,⁴ a transcription, a critical apparatus and a set of meta-data which can either describe the physical document (date of writing, place of dispatch, sender, recipient, etc.) or describe its content (scientific topics discussed, quoted persons, etc.). In addition to the Omeka S infrastructure, Semantic Web technologies have been used to offer advanced tools for the exploitation of this corpus. For this purpose, the RDF model is used to describe facts, the RDFS language is used to represent the domain knowledge and the SPARQL query language allows the interrogation of the corpus graph. When it comes to the exploration of an historical corpus, one of the main issues is to be able to put forward new and unexpected relations between individuals, institutions, scientific works, etc. This article introduces a navigation tool which has been developed for the correspondence corpus exploration by exploiting the similarities between documents. This system relies on the use of a SPARQL query transformation rule mechanism which can help to highlight unexpected relations between resources.

The remainder of this article is organized as follows. First, some background notions about Semantic Web technologies are given (Section 2). A discussion about the visualization and the exploration of cultural heritage data is provided afterward (Section 3). This section presents some related works before focusing on the case of the Henri Poincaré correspondence corpus. Through an example, the navigation tool developed for the exploration of the corpus is presented (Section 4). A discussion related to this system architecture and reusability is provided (Section 5). Section 6 concludes and points out some future works.

2 Preliminaries on Semantic Web Technologies

This section introduces the Semantic Web technologies used to represent and to query correspondence corpus data. All of these technologies are defined as recommendations of the World Wide Web Consortium (W3C).

³ <http://henripoincare.fr>.

⁴ Some are unavailable due to copyright rules.

Resource Description Framework (RDF) [10] is a knowledge representation model which is based on the definition of statements, called *triples*, of the form $\langle \text{subject} \text{ predicate } \text{object} \rangle$. A set of triples forms an RDF graph, which is a directed labeled graph. An RDF graph is composed of three kinds of nodes: named resources, which are described using an Internationalized Resource Identifier (IRI), anonymous resources which represent resources not explicitly identified and literals which correspond to constant values of a specific datatype (e.g. `string`, `integer`, `date`). The nodes of an RDF graph are linked by the use of properties. As an example, the triple $\langle \text{letter11} \text{ sentBy } \text{charlesHermite} \rangle$ states that Charles Hermite is the sender of `letter11`.

RDF Schema (RDFS) [4] is a knowledge representation language extending the RDF model. In particular, it allows to introduce hierarchies between classes and properties with the use of the `rdfs:subClassOf` and `rdfs:subPropertyOf` properties. The RDFS deduction refers to the application of inference rules for extracting knowledge from an existing knowledge base. There are other knowledge representation languages available for RDF, such as OWL, which are not considered in this article.

SPARQL is a language used to query and to update RDF graphs [16]. The SPARQL queries considered in this document have the following form: a `SELECT` clause, containing one or several anonymous resources, followed by a `WHERE` clause, composed of a graph pattern divided into one or several triple patterns, and a possible `FILTER` clause that is mainly used to add constraints for the literal values associated to the properties. Consider the following informal query:

$$Q = \left| \begin{array}{l} \text{Give me the letters sent by Henri Poincaré between 1890} \\ \text{and 1900 and having abelian functions as a topic.} \end{array} \right.$$

This query can be expressed using SPARQL:

$$Q = \left| \begin{array}{l} \text{SELECT ?l} \\ \text{WHERE \{ } \\ \quad \text{?l a letter .} \\ \quad \text{?l sentBy henriPoincaré .} \\ \quad \text{?l hasSubject abelianFunctions .} \\ \quad \text{?l hasWritingDate ?y .} \\ \quad \text{FILTER(YEAR(?y) >= 1890} \\ \quad \quad \text{AND YEAR(?y) <= 1900)} \\ \text{\} } \end{array} \right.$$

For the remainder of the document, queries are presented in an informal way though all of them correspond to actual SPARQL queries.

3 Exploration of a History of Science Corpus

3.1 Related work

Semantic Web technologies have been widely used to represent and to exploit cultural heritage corpora. Many institutions (museums, libraries, archives, etc.) use

Semantic Web technologies to structure and to share resources. Some important initiatives can be mentioned such as the *Digital Public Library of America*, which shares a collection of free of charge books, and the *Europeana* platform [8], which provides access to resources from cultural institutions of the European Union.

Several research projects aim at providing researchers with great visualization and exploration portals. A series of tools have been developed in the context of the *CultureSampo* system which is dedicated to the use of Semantic Web for the valorization of Finnish culture [9]. The Sampo model includes a collaborative publication system as well as different visualization and exploration portals. These tools benefit from data type diversity in order to facilitate item browsing. As an example, a portal provides a map browsing environment which shows items associated with places. Another tool proposes to navigate through a timeline which shows some historical events related to Finnish history. SCRABS (Smart Context-awaRe Browsing assistant for cultural EnvironmentS) [1] is a system which allows context-driven browsing of cultural environments and which is an extension of some preliminary work proposed by Colace et al [7]. For more information related to visualization of cultural heritage data, please see [18] which provides a great overview of existing approaches and presents some future challenges. Beyond the visualization question, the use of Semantic Web technologies for historical research brought a lot of issues and challenges related to data creation, editing, querying, analysis, etc. [12].

3.2 The Need of a Navigation Tool for the Henri Poincaré Correspondence Corpus

The Henri Poincaré correspondence corpus comprises letters from 400 correspondents that refer to 1600 quoted peoples. This set of individuals includes a lot of French and foreign scientists, some members of various institutions and learned societies as well as colleagues, friends, acquaintances and family relations of Poincaré. The letters of the corpus are associated with various topics which can be related to scientific contents (e.g. abelian functions, three-body problem), to institutions and learned societies (e.g. *Bureau des longitudes*, *École polytechnique*) or be some topics related to the life of Henri Poincaré (e.g. study trip to Norway and Sweden, family and friend visits).

One way to access corpus data is to use SPARQL queries to retrieve resources matching a set of conditions. However, this approach leads to three difficulties. First, this requires a good knowledge of SPARQL syntax which is not suitable to all users. Another issue is that it is necessary to have a good understanding of the ontology schema in order to write adapted queries. Finally, even if users are able to write queries or to generate them through a form-based system, their executions return exact results which sometimes does not encourage the exploration aspect when studying the corpus. In this historical context, it would be more suitable to use a dynamic tool which proposes a visual exploration of the corpus. It would also be in line with a vulgarization effort related to this historical corpus.

Navigation can be a good option for exploring a corpus. For the Henri Poincaré correspondence corpus, the idea would be to start from an initial letter and then to look for similar resources: the letters sent during the same temporal period, the letters sent by or to the same correspondent or the same institution or the letters having one or several related topics, etc. In some situations, most of the related resources could already be known to historians, but in others situations, the tool can highlight unexpected relations between resources and be a starting point for the exploration of new historical perspectives. As an example, these new relations can reveal some patterns between scientific topics and individuals (as correspondents or quoted people), in particular for actors considered as minors for whom no research work has been initiated. That is why creating such a navigation tool seemed to be an interesting option for this corpus.

4 A Navigation Tool for Exploring Corpora

This section presents a tool which has been developed to navigate through the Henri Poincaré correspondence corpus. This system can be used for the exploration of any Semantic Web graph. It is particularly relevant for the exploration of historical corpora because results are presented within a chronological-based interface. A demonstration video of this tool is available online.⁵ In this section, a use case related to the Henri Poincaré correspondence corpus is presented.

4.1 Overview of the Tool

The tool is available as a Web interface which allows users to generate SPARQL queries and to visualize, to filter and to export results. This interface is divided into four main blocks as shown by the excerpt presented in Figure 1.

The top-left block gives some information about the initial resource related to the current search process. In our example, this resource corresponds to a letter sent by Henri Poincaré to Gösta Mittag-Leffler on June 29, 1881. The block presents the IRI associated to the resource as well as its label. Clicking on one of these elements redirects on a human-readable representation of the resource.⁶ In our example, this redirects to an HTML page⁷ giving information about the letter: a scan of the original document, a transcription, a set of meta-data and a critical apparatus. The info box on the top-right corner provides a description of the resource based on the `rdfs:comment` property value.

The "update" button opens an advanced search window in which one can retrieve a resource from the RDF graph with a search based on its type and its label. Once the selection of a resource has been validated, a new application process is initialized centered around this resource.

⁵ <https://videos.ahp-numerique.fr/videos/watch/f90ff003-39db-4b4c-ade6-fb18b86d9244>.

⁶ This may depend on the RDF graph used with the tool. In some situations, no IRI dereferencing mechanism is provided for resources.

⁷ <http://henripoincare.fr/s/correspondance/item/5834>.

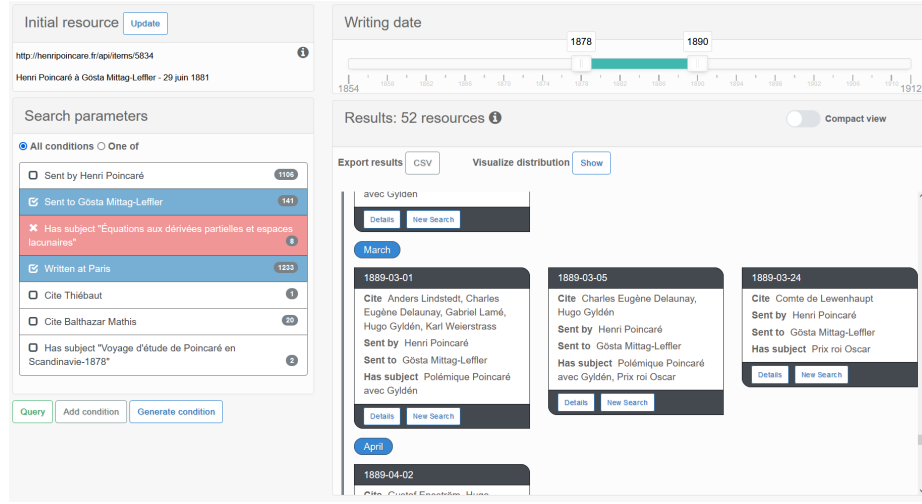


Fig. 1. The navigation tool interface in use for the exploration of the Henri Poincaré correspondence corpus.

The bottom-left block gathers a set of search conditions which can be used to create SPARQL queries. These conditions are generated based on the initial resource and are presented in a human-readable form by reusing the labels associated with the ontology properties and with the graph resources. For the running example, this leads to the generation of seven conditions related to the sender, the recipient, the topics and the persons quoted. The properties that must be used by the tool to generate conditions are specified in a configuration file. Other meta-data exist related to letters but only those relevant for this use case have been selected. As an example, the property which specifies the number of pages of the original document has not been identified as relevant and is not considered by the tool when generating conditions, though this property could easily be added by editing the configuration file. Next to each condition, a number indicates the resources matching the given condition on the RDF graph. Users can select different conditions by clicking on them. A first click makes the condition panel background blue, checks the box on the left, and means that the system considers resources matching the condition. A second click makes the background red, inserts a cross in the box, and means that the system considers resources which do not match this condition. Another click puts the condition in its initial state, meaning that it is not considered by the system for the search over the RDF graph. Users can choose to look for resources matching all conditions ("All conditions" mode) or at least one of them ("One of" mode).

By clicking on the "query" button, the system updates the results presented on the bottom-right block of the interface. For the running example, the conditions lead to the formulation of the following query:

$$Q_{search} = \left| \begin{array}{l} \text{Give me the letters sent to Gösta Mittag-Leffler,} \\ \text{written at Paris, and which do not have } \acute{E}quations \\ \text{aux d\'eriv\'ees partielles et espaces lacunaires as a topic.} \end{array} \right.$$

The result block presents a set of resources in a chronological-based view. For each result, some information about the resource is given. For the Henri Poincaré correspondence corpus, each letter is described with its label, its sender, its recipient, the topics and the persons quoted. These properties are configurable and can be different from the properties used to generate conditions. Clicking on the resource box redirects on a human-readable representation of the resource. The system proposes to export the results in a CSV file which embeds the IRI and some information about each resource. Another functionality concerns the presentation of a bar chart which expresses the distribution of the results in relation to the chosen date property. Figure 2 presents the distribution related to the letters sent by Henri Poincaré to Gösta Mittag-Leffler. One can identify the temporal periods during which the correspondence with Gösta Mittag-Leffler seems to be more or less active.

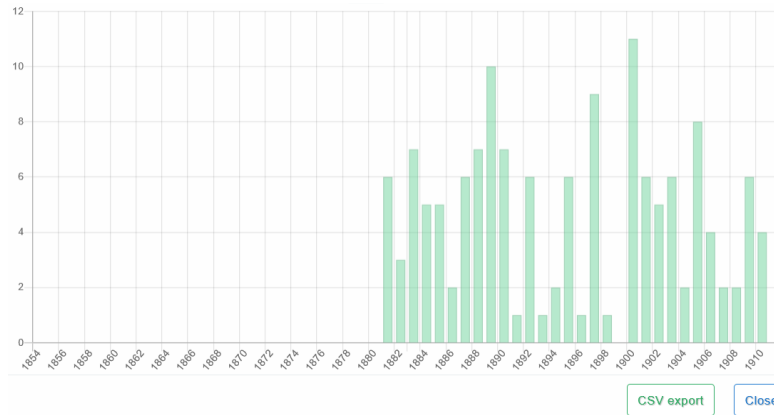


Fig. 2. The distribution of letters sent by Henri Poincaré to Gösta Mittag-Leffler.

It is also possible to start a new search process centered around one of the letters presented in the result block. The idea of the system is to start with an initial resource and to explore the corpus by navigating from a resource to another by taking different paths. This way of exploring the corpus could lead to the identification of unexpected relations between the elements of the corpus. It can be a way to put forward actors and works generally considered to be secondary.

Above the result block, a date slider can be used to filter the set of presented results. This filtering tool is related to a configurable date property. For this use

case, this property corresponds to the letter writing date and can be set between 1854 and 1912, which correspond to Henri Poincaré's life dates. The initial range of the slider is centered around the writing date of the selected letter.

4.2 Going Further by Applying a SPARQL Query Transformation Mechanism

In some situations, the generated conditions may not be sufficient for providing interesting or surprising results. A first solution, which is included in the tool, is to let users manually add a new condition. For this purpose, the "Add manual condition" button shows a new window in which users can create conditions by selecting one of the available properties and by attributing a value that can be retrieved from a list or manually entered if necessary. This functionality may be useful in several situations but it can be a long process to find and select the appropriate properties and values. Moreover, adding a condition which is not at all related to the initial resource would not have a lot of sense in a navigation tool that relies on resource similarities.

Another idea to overcome this issue is to be able to generate new conditions that the initial resource would not necessary match but which are related to its characteristics. For this purpose, the system relies on the use of transformation rules that can be applied to provide new filtering conditions which are related to the conditions generated from the initial resource characteristics. These transformation rules are defined and applied using the SQTRL (SPARQL Query Transformation Rule Language) tool, which has been introduced to allow flexible querying with the SPARQL language [5]. An SQTRL rule r is defined by a set of fields:

- **name**(r): a string identifying the rule;
- **context**(r): an RDF triple pattern to search for in the database;
- **left**(r): an RDF triple pattern to be matched with the initial query;
- **right**(r): an RDF triple pattern which replaces **left**(r);
- **cost**(r): a positive number representing the transformation cost;
- **explanation**(r): a text describing the rule in an informal way and which may contain variables from **context**, **left** and **right**.

Let \mathcal{G} be an RDF graph, Q be a SPARQL query, and r be an SQTRL rule. r can be applied to Q if **context**(r) can be bound with \mathcal{G} and **left**(r) can be bound with the body of Q . The application of r to Q generates Q' by substituting **left**(r) by **right**(r). A cost is associated with each of the rules to favor the application of certain rules. Rules can be application-independent (generalization, specialization, etc.) or application-dependent (exchange of the sender and recipient of the letter, replacement of the topic of the letter by a related topic, etc.). The rules are represented using an XML syntax. Figure 3 shows an example of a rule which is used to replace an instance ($?o$) by a variable ($?x$). This variable is of the type of one of the classes to which the instance belongs ($?C$). An application of this rule can lead to the transformation of the informal query Q_0 into Q_1 .

```

<rule name="Generalize instance in object position">
  <context>?o a ?C</context>
  <left>?s ?p ?o</left>
  <right>?s ?p ?x . ?x a ?C</right>
  <cost>2.0</cost>
  <explanation>
    Generalize ?o in any instance of ?C.
  </explanation>
</rule>

```

Fig. 3. An example of SQTRL rule in XML syntax.

$$Q_0 = \left| \begin{array}{l} \text{Give me the letters sent} \\ \text{to Gösta Mittag-Leffler.} \end{array} \right. \quad Q_1 = \left| \begin{array}{l} \text{Give me the letters sent} \\ \text{to a mathematician.} \end{array} \right.$$

The definition of transformation rules is a knowledge acquisition task involving an application domain expert and a knowledge engineer. Some details about this transformation rule mechanism as well as some application examples can be found in [5, 6]. In the context of this navigation system, the idea is to reuse and to adapt this mechanism to generate new conditions related to the current search process. The set of conditions used to create search over the corpus can be expressed within the body of a SPARQL query. This initial SPARQL query can be set as the entry of the SPARQL query transformation rule mechanism. The navigation system can then extract new conditions from the transformed bodies of the generated SPARQL queries.

On the user interface, the "More" button is used to generate new conditions in an iterative way. For the running example, the first click on the button generates two new conditions: {sent to Henri Poincaré} and {sent by Gösta Mittag-Leffler}. This is related to the application of a transformation rule which exchanges the sender and the recipient of a letter. Another action of the button adds the condition {has topic Mathematics}, based on the application of a topic generalization rule which replaces *Équations aux dérivées partielles et espaces lacunaires* by *Mathematics*. By applying the same transformation rule, the tool generates the condition {has topic Travel}. Two other rule applications generate conditions to search for letters sent of received by one of the persons quoted: {has for correspondent Thiébaud} and {has for correspondent Balthazar Mathis}.

5 System Architecture and Reusability

5.1 Technical Architecture

The system is decomposed into two distinct applications: a Web user interface defined using the HTML, CSS and Javascript technologies which manages user interactions as well as the visualization and the export of results, and a Java application which uses the Jena API [11] to extract knowledge from RDF graphs

through SPARQL queries. The Java application works as an interface between user's actions and the RDF graph by generating the appropriate SPARQL queries and by returning the results with the desired structure. The two applications communicate by the exposition and the call of Web services following the REST architecture.⁸ As an example, a service is dedicated to the retrieval of application properties, another service retrieves details about the initial resource, a service manages the labels associated with properties and resources, a service retrieves the results corresponding to the conditions checked by users, etc.

The system and its source code are available online on a GitHub repository.⁹ On this repository, on the "app" folder is available the Java application, packaged as a runnable Jar file, and the Web application which is composed of several source files (.html, .css and .js). Both parts are required to make the application functional. In the "source" folder, one can retrieve the full application source code which is not required for running the tool—see the documentation available on the GitHub repository for more technical details.

5.2 System Reusability

When developing this tool, one of the main challenges was to ensure its reusability with other corpora. In this context, the Java application comes with a specific parameter file which can be used to define several values: the URL of the SPARQL endpoint; the path to the transformation rule file;¹⁰ the list of properties to be used for condition generation; the list of properties to be displayed for each result; the property and language for labels; the date property and the temporal interval associated with results filtering.

The documentation available on the GitHub repository describes another use case which is related to the search of literary works by querying the *DBpedia* public SPARQL endpoint.¹¹ DBpedia is a community effort to extract and to structure content from information created in various Wikimedia projects [2]. The Wikimedia movement aims at providing educational content to the world through diverse initiatives and projects whose best known is Wikipedia. In this situation, the navigation tool can be configured with various properties in order to find similar literary works: the themes covered, the literary genres, the author, the author's nationality, the artistic movements associated with the author, etc. The publication date can be defined as the date property to filter results. Through the chronological-based view, one can easily navigate through results and access to works of interest published in a temporal period relatively close to the publication date of the initial work.

⁸ REST (*REpresentational State Transfer*) characterizes a software architecture style which can be used to create a communication bridge between software components with the use of HTTP methods (GET, POST, PUT, etc.).

⁹ https://github.com/nlasolle/rdf_navigation_tool.

¹⁰ An application-independent transformation rule file is available on the repository.

¹¹ <https://dbpedia.org/sparql/>.

6 Conclusion

The Henri Poincaré correspondence corpus is a historical corpus that is of interest for many scientific, societal and institutional aspects. Semantic Web technologies have been used to represent corpus data and to create advanced search tools. A navigation system, accessible through a Web user interface, has been proposed for exploring the Henri Poincaré correspondence corpus. This system benefits from the use of a SPARQL query transformation mechanism for exploiting the relations between the elements of the corpus. It is a generic tool for Semantic Web graphs exploration and could thus be reused with other corpora.

Future work is considered for improving this system. Some improvements concern adding new features to the user interface. One idea is to be able to remove a condition from the list if it does not present any interest for the current navigation. Another idea is to be able to select a subset of the results before exporting data to CSV. This export could also be proposed in other formats (JSON, RDF, etc.). A major improvement would be to keep a trace of any action performed with the tool and thus being able to save the research process. In the context of a historical corpus, the research methodology is an important aspect of the work associated with results presentation. Another idea is to provide the user with an explanation of conditions generated by the use of the SPARQL query transformation mechanism. This could be the description of which transformation rule has been applied and for which resources. Finally, we would like to make this tool available from the Henri Poincaré correspondence website. The idea is to provide a link accessible from a resource dedicated Web page and whose action would be to start a navigation process centered around this resource.

Acknowledgement. This work is supported partly by the French PIA project “Lorraine Université d’Excellence”, reference ANR-15-IDEX-04-LUE. We greatly thank Pierre Willaime for his attentive reading of the user documentation.

References

1. Amato, F., Moscato, V., Picariello, A., Colace, F., Santo, M.D., Schreiber, F.A., Tanca, L.: Big Data Meets Digital Cultural Heritage: Design and Implementation of SCRABS, A Smart Context-AwaRe Browsing Assistant for Cultural EnvironmentS. *Journal on Computing and Cultural Heritage* **10**(1) (April 2017). <https://doi.org/10.1145/3012286>
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. In: Aberer, K., Choi, K.S., Noy, N., Allemang, D., Lee, K.I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) *The Semantic Web*. pp. 722–735. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
3. Boulaire, C., Carabelli, R.: Du digital naïve au bricoleur numérique : les images et le logiciel Omeka. In: Cavalié, É., Clavert, F., Legendre, O., Martin, D. (eds.) *Expérimenter les humanités numériques. Des outils individuels aux projets collectifs*, chap. 7, pp. 81–103. Les Presses de l’Université de Montréal, Montréal, Québec (2017)

4. Brickley, D., Guha, R.V.: RDF Schema 1.1, W3C recommendation (2014), <https://www.w3.org/TR/rdf-schema/>, last consultation: April 2021
5. Bruneau, O., Gaillard, E., Lasolle, N., Lieber, J., Nauer, E., Reynaud, J.: A SPARQL Query Transformation Rule Language — Application to Retrieval and Adaptation in Case-Based Reasoning. In: Aha, D., Lieber, J. (eds.) *Case-Based Reasoning Research and Development. ICCBR 2017*. pp. 76–91. *Lecture Notes in Computer Science*, Springer, Cham (2017)
6. Bruneau, O., Lasolle, N., Lieber, J., Nauer, E., Pavlova, S., Rollet, L.: Applying and Developing Semantic Web Technologies for Exploiting a Corpus in History of Science: the Case Study of the Henri Poincaré Correspondence. *Semantic Web – Interoperability, Usability, Applicability* **12**(2), 359–378 (2021)
7. Colace, F., De Santo, M., Moscato, V., Picariello, A., Schreiber, F.A., Tanca, L.: PATCH: A Portable Context-Aware ATlas for Browsing Cultural Heritage, pp. 345–361. *Springer International Publishing*, Cham (2015). https://doi.org/10.1007/978-3-319-20062-0_17
8. Haslhofer, B., Isaac, A.: data.europeana.eu: The europeana linked open data pilot. In: Baker, T., Ileana Hillmann, D., Isaac, A. (eds.) *International Conference on Dublin Core and Metadata Applications*. pp. 94–104 (2011)
9. Hyvönen, E., Mäkelä, E., Kauppinen, T., Alm, O., Kurki, J., Ruotsalo, T., Seppälä, K., Takala, J., Puputti, K., Kuittinen, H., Viljanen, K., Tuominen, J., Palonen, T., Frosterus, M., Sinkkilä, R., Paakkari, P., Laitio, J., Nyberg, K.: CultureSampo - Finnish Culture on the Semantic Web 2.0. Thematic Perspectives for the End-user. In: *Museums and the Web 2009: Proceedings. Archives & Museum Informatics* (2009)
10. Manola, F., Miller, E., McBride, B., Schreiber, G., Raimond, Y.: RDF 1.1 Primer, W3C Recommendation (2014), <https://www.w3.org/TR/rdf11-primer/>, last consultation: April 2021
11. McBride, B.: Jena: a Semantic Web toolkit. *IEEE Internet Computing* **6**(6), 55–59 (2002). <https://doi.org/10.1109/MIC.2002.1067737>
12. Meroño-Peñuela, A., Ashkpour, A., Van Erp, M., Mandemakers, K., Breure, L., Scharnhorst, A., Schlobach, S., Van Harmelen, F.: Semantic Technologies for Historical Research: A Survey. *Semantic Web – Interoperability, Usability, Applicability* **6**(6), 539–564 (2015)
13. Nabonnand, P. (ed.): *La correspondance entre Henri Poincaré et Gösta Mittag-Leffler*. Birkhäuser, Basel (1998)
14. Poincaré, H.: Sur la théorie des quanta. *Journal de physique théorique et appliquée* **2**(1), 5–34 (1912)
15. Poincaré, H.: *La Science et l’Hypothèse*. Flammarion (1902)
16. Prud’hommeaux, E.: SPARQL Query Language for RDF, W3C Recommendation (2008), <http://www.w3.org/TR/rdf-sparql-query/>, last consultation: April 2021
17. Rollet, L. (ed.): *La correspondance de jeunesse d’Henri Poincaré : les années de formation. De l’École polytechnique à l’École des Mines (1873-1878)*. *Publications of the Henri Poincaré Archives*, Birkhäuser (2017). <https://doi.org/10.1007/978-3-319-55959-9>
18. Windhager, F., Federico, P., Schreder, G., Glinka, K., Dörk, M., Miksch, S., Mayr, E.: Visualization of Cultural Heritage Collection Data: State of the Art and Future Challenges. *IEEE Transactions on visualization and Computer Graphics* **25**(6), 2311–2330 (2018). <https://doi.org/10.1109/TVCG.2018.2830759>