Harold: an iterative and interactive query system for exploring cultural heritage corpus

Prunelle D. Treuil[1,*], Olivier Bruneau[2], Jean Lieber[1], Emmanuel Nauer[1] and Laurent Rollet[2]

[1]LORIA CNRS/INRIA/Université de Lorraine, France
[2]AHP-PReST, Université de Lorraine, Université de Strasbourg, France

Abstract

With the development of Semantic Web technologies in digital humanities, more and more users with little knowledge about these technologies need to interact with them. This paper presents Harold, a system for accessing the cultural heritage corpus without having to know any particular computer language, such as SPARQL. Harold is a conversational system with which a historian can interact using a user-friendly query interface together with navigation access to explore the corpus. Harold organizes the documents into a hierarchical structure using formal concept analysis (FCA) to provide a synthetic way to navigate the documents. The user may interact with the hierarchy concepts to focus on relevant documents and remove irrelevant ones. In addition, an ontology management interface is provided to assist the user in managing concepts related to their research problem. This ontology can be used by the retrieval process to better structure hierarchical access to the documents. Moreover, the concepts built by FCA provide interesting information that can guide the user to a new retrieval step to find more relevant documents related to their research problem.

Keywords

Ontology management, Conversational system, Formal Concept Analysis, Cultural Heritage Collection

1. Introduction

This paper presents *Harold*, a conversational system that helps to explore textual corpora, such as the correspondence of Henri Poincaré. Currently developed in collaboration with the Henri Poincaré Archives, the system enables interactive exploration of the digitized and annotated correspondence using the Archives' SPARQL endpoint. The Henri Poincaré Archives, a center for the history and philosophy of science, holds parts of the personal papers of various mathematicians and computer scientists. This includes more than $2,000$ letters exchanged by Henri Poincaré, now digitized, manually transcribed and enriched with metadata by the historians of the Archives. A key figure in mathematics, physics and philosophy of science, the Henri Poincaré (1854-1912) correspondence offers valuable insight into his scientific activities and exchanges with fellow researchers of his time. Although the SPARQL endpoint supports advanced queries into the knowledge graph and ontologies developed at the Archives, it remains a challenge for non-experts, hence the need for an accessible interface like Harold. Even if Harold is originally developed for the Henri Poincaré Archives, it will be tested on other corpora as researchers working on other digital humanities corpora have expressed interest in this tool. This would allow us to test Harold on new ontologies and knowledge graphs with different vocabularies.

To simplify access to a corpus without requiring SPARQL queries, the Harold system offers a more user-friendly interface. This work addresses three key aspects: (1) Historical research questions are often vague and cannot be answered with a single query, for example "What has Henri Poincaré exchanged about mathematics for physics?". Harold supports an iterative interactive exploration using FCA [1], allowing users to refine their search by grouping letters according to shared properties. (2) Beyond basic metadata (sender, recipient, date, etc.), the content of the letters is crucial. NLP techniques are used to extract significant terms (e.g. single terms, nominal groups, named entities) from the full text. These terms serve as additional semantic properties for exploration. (3) Historical research also involves domain

SemDH 2025. Second International Workshop of Semantic Digital Humanities
*Corresponding author.
prunelle.daudre--treuil@loria.fr (P. D. Treuil); olivier.bruneau@univ-lorraine.fr (O. Bruneau); jean.lieber@loria.fr (J. Lieber); emmanuel.nauer@loria.fr (E. Nauer); laurent.rollet@univ-lorraine.fr (L. Rollet)

knowledge. Harold lets users build and manage an ontology, using NLP-extracted terms as candidate concepts. Harold utilizes the ontology to generalize term annotations by associating letters with broader conceptual categories. This enhances both the hierarchical structuring of the corpus and the retrieval process, allowing queries on general terms to also return letters containing more specific related terms. Harold uses the ontology to associate letters with broader terms, improving both hierarchical organization and retrieval by gathering letters using more general concepts or more specific concepts. For example, if the user has indicated that the *physics* concept is more general than the *heat propagation* concept, then every time the letters about physics are searched, the ones about heat propagation can also be retrieved. This dynamic interaction between the user, the corpus, and the ontology supports knowledge discovery.l research problem involves knowledge about the research topic.

The paper is organized as follows. Section 2 presents the related work: solutions allowing non-specialists to interact with ontologies, as well as the use of FCA in document access. Section 3 describes preliminaries: the letters of the Henri Poincaré correspondence, the general ontology used to annotate the letters, and the FCA theory. Section 4 presents the Harold system: the NLP extraction process to enrich the letter annotations by their content and shows the main functionalities of the Harold system illustrated by a running example. Section 5 describes the ontology management tool of the system.

2. Related work

Numerous works address the problem of interrogating knowledge base, especially represented in RDF or accessible through a SPARQL endpoint. A classification of approaches that address this problem, from the most informal (natural language) to the most formal (SPARQL), can be found in [2]. Some works focus on the transformation of the problem, given in natural language, into SPARQL queries. Diefenbach et al. [3] present a survey on systems that address question answering (QA) problems. The question, given in natural language, has to be solved by searching the answer in an RDF knowledge base. These systems use different NLP techniques, most of the time named entity recognition, part-of-speech tagging, and dependency analysis to translate a natural language query into a SPARQL query. For example, in the SWIP system [4], a first step consists of identifying terms in the question related to classes and instances of the knowledge base and, according to the query patterns, generating a SPARQL query. More recent works have focused on using deep learning QA techniques, mainly Large Language Models (LLMs) to answer questions directly written in natural language. Biancofiore et al. [5] propose a survey on interactive QA and Zaib et al. [6], a survey on conversational QA. Both highlight that QA in several steps allows the system to better understand the question in natural language and its context. Finally, Lan et al. [7] different QA techniques on knowledge bases with queries in natural language.

As query patterns are represented in an ontology, the transformation from a question in natural language to a SPARQL query also uses SPARQL for mapping some part of the text of the question with knowledge base resources. Other works propose to use specific user interfaces to query knowledge bases without directly writing a SPARQL query. For example, Sparklis proposes an interface that assists the user in constructing the SPARQL query [8]. In Sparklis, the construction is based on forms in which the user will progressively select classes, relations, and values from a selected knowledge base. This selection is transformed both into an SPARQL query and, to give feedback to the user, in natural language, to be sure that the interaction with the system has conduct to the creation of a SPARQL query corresponding to the user question.

Other systems propose to use a visual query language. In Nitelight, for example, the query takes the form of a graph and the user has to construct the graph corresponding to the query [9]. Finally, some works propose specific interfaces dedicated to the knowledge base to query, in particular, the works already done on the exploitation of the Henri Poincaré correspondence. In addition to proposing a dedicated form to query and navigate the correspondence, other search functionalities are available, such as a similarity search based on letter metadata and generalization rules [10], and approximate search by transformation of SPARQL queries to produce new SPARQL queries close to the initial query of the user [11].

Another point addressed in this work relies on a synthetic access instead of an individual letter access, and the fact that a research problem required many steps and therefore interaction with the search system. The work presented in this paper is inspired by the CreChainDo system [12], an iterative and interactive information retrieval system that organizes GOOGLE web answers using formal concept analysis (FCA). As FCA provides a natural way to organize objects according to their properties (see Section 3.2), FCA is used to organize answers provided by a search engine in a hierarchy. Navigation in the hierarchy helps the user explore a structured and synthetic result, and the user can interact with the result to indicate concepts that are relevant or irrelevant for a given information retrieval task. These user choices have an impact on the hierarchy: some concepts are removed if they are irrelevant, and new ones appear by triggering additional web queries about relevant concepts.

Finally, the ontology building process presented in this paper is inspired by some of the many approaches that can be found in the literature (see [13] for a survey of ontology learning techniques). The ontology building is helped by candidate concepts resulting from an NLP process associated with the use of FCA to highlight frequent terms. So, it addresses four layers of the *"Ontology Learning Layer Cake"* proposed in [14]: identifying terms, identifying synonyms (this is not detailed in this paper), identifying concepts, and organizing concepts into a concept hierarchy. The ontology building process proposed in this work also relies on [15] for the interactive and interactive approach. Even if our objective in this work is not to build concept definitions as in [15], we share the idea of progressively building the ontology thanks to a sequence of steps guided by the domain expert who provides additional useful knowledge, in a given step, to find other relevant knowledge pieces (new concepts related to their research problem) in the next step.

3. Preliminaries

3.1. The Henri Poincaré corpus and ontology

The Henri Poincaré correspondence corpus includes more than 2000 letters exchanged with more than 300 correspondents in five languages (mainly French with sporadic use of English, German, Sami, and Swedish), written between his entry to the *École polytechnique* in 1873 and his death in 1912. These documents have been digitized, manually transcribed, and semantically annotated with general metadata and contextual footnotes, forming the corpus's *critical apparatus*, through over 30 years of historical work.

The letters are available in three formats: thematic printed volumes [16], a website[1] using the content management system for cultural heritage collection Omeka S [17], and a SPARQL endpoint (login required). The website offers a basic keyword search over raw text, without support for complex queries. The SPARQL endpoint enables precise access to the knowledge base but requires familiarity with SPARQL and its underlying vocabulary. The Henri Poincaré ontology, developed by the Henri Poincaré Archives, organizes all metadata related to the correspondence. It includes properties for describing the letters, correspondents, archival information (e.g. storage, copyrights) and publications (e.g., books, articles). If the ontology also covers other documents related to Poincaré's life, such as scientific articles, books, and theses, only his letters are used by Harold.

Each letter is described by semantic annotations using RDF triples, as illustrated in Fig. 1. The triples use properties of the Henri Poincaré ontology (`ahpo` prefix), such as the sender (`ahpo:sentBy`), the recipient (`ahpo:sentTo`), the language (`ahpo:language`), the date (`ahpo:writingDate`), and the written place (`ahpo:writtenAt`).

3.2. Formal concept analysis

FCA is a mathematical approach to data analysis based on lattice theory. A *formal context* is a triple $\mathcal{K} = (O, A, R)$, where O is a set of individuals (called *objects*), A is a set of properties (called *attributes*)

[1] https://henripoincare.fr/

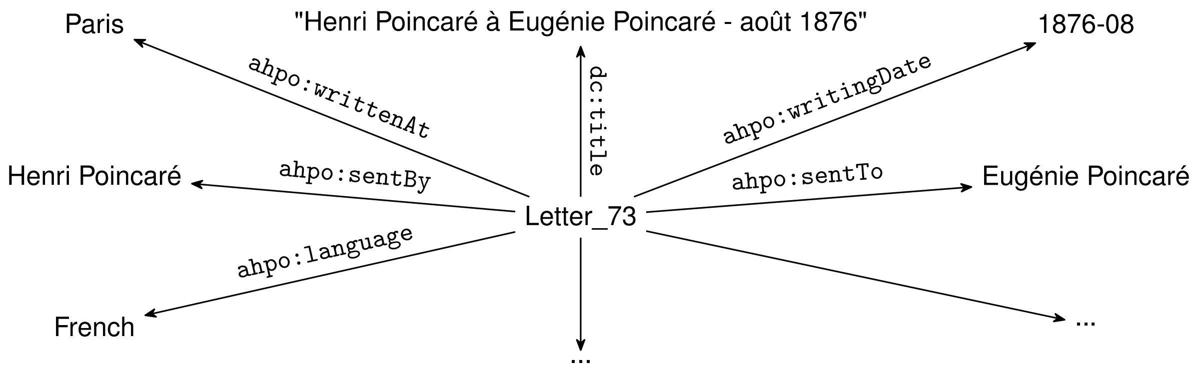
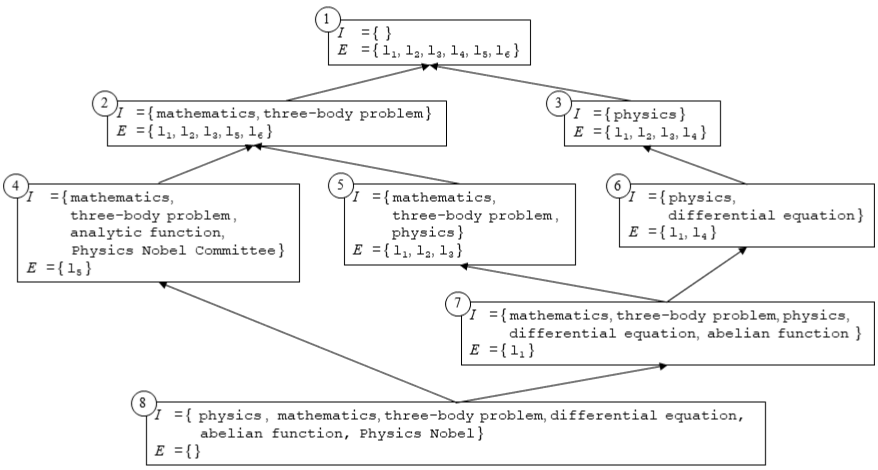


Figure 1: RDF triples associated to the letter *"Henri Poincaré to Eugénie Poincaré - August 1876."*

	physics	mathematics	three − body problem	differential equation	abelian function	analitic function	Physics Nobel Commitee
l_1	×	×	×	×	×		
l_2	×	×	×				
l_3	×	×	×				
l_4	×			×			
l_5		×	×			×	×
l_6		×	×				

(a)



(b)

Figure 2: A binary context (a) and the corresponding concept lattice (b).

and R is the relation on $O \times A$ stating that an object is described by a property [1]. Table of Fig. 2 (a) gives an example of context: O is a set of 6 letters ($\mathtt{l}_1$,...,$\mathtt{l}_6$) and A is the set of properties composed of 6 terms and 1 recipient annotating the letters. A *formal concept* is a pair (I, E), where E is a maximal set of individuals (called *extent*) and I is a maximal set of properties (called *intent*) shared by this extent. For example, $(\{\mathtt{physics}, \mathtt{differential\ equation}\}, \{\mathtt{l}_1, \mathtt{l}_4\})$ is a concept.

Furthermore, the set $\mathcal{C}_{\mathcal{K}}$ of all formal concepts of the context $\mathcal{K} = (O, A, R)$ is partially ordered by extent inclusion, also called *specialization* (denoted by $\leq_{\mathcal{K}}$) between concepts. $\mathcal{L} = \langle \mathcal{C}_{\mathcal{K}}, \leq_{\mathcal{K}} \rangle$ is a complete lattice, called the *concept lattice*. The lattice $\mathcal{L}$ can be drawn as a Hasse diagram where the nodes are concepts and the arrows are specialization links. Fig. 2 illustrates a binary context (a) and its corresponding lattice (b). The top concept contains all the letters; its intent is empty because there is no common property shared by all the letters. In contrast, the bottom concept is defined by the set of all properties, and its extent is empty because none of the letters is described by all the properties. Several algorithms have been proposed for the construction of concept lattices; see [1]. For Harold, CORON is used,[2] where CORON is a software platform that implements a wide set of algorithmic methods for symbolic data mining, including concept lattice construction algorithms [18].

[2]http://coron.loria.fr

4. The Harold system

The Harold system helps historians explore the Henri Poincaré correspondence by retrieving letters, analyzing their content, and acquiring knowledge. An NLP step for term identification in full texts enables the exploitation of both the letters' content and their critical apparatus. Section 4.1 details the annotation process, while Section 4.2 introduces a running example, followed by descriptions and illustrations of Harold's functionalities.

4.1. Exploiting full texts using NLP

Historians analyzing the Henri Poincaré correspondence require analysis of both the full text and the critical apparatus, not just basic metadata. To enable this, terms were extracted from each letter and its critical apparatus using both a preexisting list of key terms and automatic noun phrase extraction. Scientific terms, including those of institutes such as Nobel committees, are primarily identified through noun phrases, while terms from Henri Poincaré 's social environment, such as polytechnic slang, contain only one word. This list, curated by historians, is crucial as many terms have context-dependent meanings (e.g., "rat" refers to being late, not the animal). In addition, a list of cultural and scientific works cited in the letters is also maintained. All the terms extracted are kept in their original language, with spelling errors and abbreviations, the historians having the possibility to indicate all the variations of a given term.

4.2. Running example

As a running example, consider studying the impact of Henri Poincaré's work on physics research. The historian can use Harold to search for letters exchanged with physicists Henri Poincaré knew or containing keywords related to mathematical physics (e.g., general terms: *mathematics*, *physics*, or more precise concepts: *differential equation*, *three-body problem*). Solving this research problem requires multiple searches and content analysis. Based on the initial results, the historian can refine the searches with more specific terms or exclude irrelevant results. As they explore the letters, some knowledge can emerge within the search scope, particularly the vocabulary (i.e. terms) and concepts. Organizing these concepts in an ontology can enhance and refine the search process.

Section 4.3 presents the search form used to explore the corpus. In Section 4.4, the way the results are displayed is explained. Section 4.5 illustrates different types of interactions with Harold and shows the benefit of an interactive and interactive exploration process.

4.3. Searching letters by properties

Harold allows to search, in the Henri Poincaré correspondence, letters that have some properties using a specific search form. This form also allows us to indicate which properties must be taken into account by the FCA using checkboxes. For example, a historian can indicate that they are interested in knowing who exchanged on a given subject (i.e., `Sender` and `Recipient` properties) and what terms appear in both the letters and their critical apparatus (`Containing` properties). The content of this form is transformed into a SPARQL query to search for letters that have the desired properties. **This query simply filters all letters with respect to the given criteria and retrieves the properties chosen by the user.** Finding all relevant letters in a single attempt is rare, requiring iterative interaction between the user and the system. Harold dynamically manages the set of letters used by FCA, which evolves by adding relevant letters through new queries or removing those irrelevant to the current research problem. In the interface, *Starting a new search* creates a new set, while *Complete the current search* adds new letters to the existing set.

4.4. Visualizing the results

Instead of simply listing the retrieved letters, Harold applies FCA to group them based on the selected properties, creating a synthetic view that highlights the shared characteristics (see Section 3.2). The resulting lattice is hierarchically displayed using depth-first search, as shown in Fig. 3. The sign ⌄

expands the groups to show subgroups, while ❯ collapses them. Each subgroup only displays properties not inherited from broader groups. For example, in Fig. 3, the 18 letters sent to *Physics Nobel Committee* also contain *physique mathématique*. The group size, set to 5, is adjustable by the user.

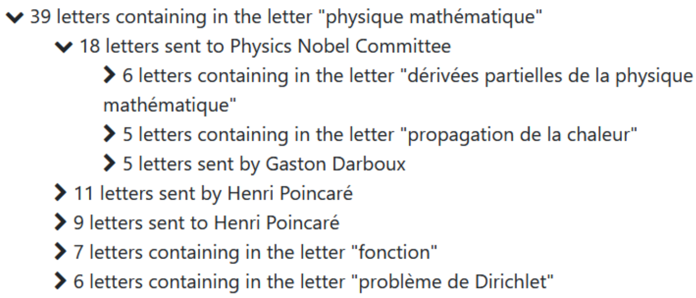


Figure 3: Hierarchical presentation of the results after starting a new search on *physique mathématique*.

In this illustration, it can be seen, from the 39 letters containing *physique mathématique*, that 18 of these letters have been sent to the *Nobel Physics Committee*. Some potentially interesting terms for the research problem also appear: *dérivées partielles de la physique mathématique* (in English, *partial derivatives of mathematical physics*), *propagation de la chaleur* (in English, *heat propagation*), and *problème de Dirichlet* (in English, *Dirichlet problem*).

4.5. Interacting with the results

To find all letters related to the topic *"Mathematics for physics"*, a search for *physique mathématique* alone is insufficient, as many letters do not contain this exact phrase; Harold's interactive features allow the user to refine searches, add relevant letters, or exclude irrelevant ones, thereby improving FCA results. As shown in Fig. 4, several actions are available for each letter group (here the 18 letters sent to the *Physics Nobel Committee*), the first three for letter analysis, then one for ontology management (see Section 5), and the last one for accessing the letters themselves.

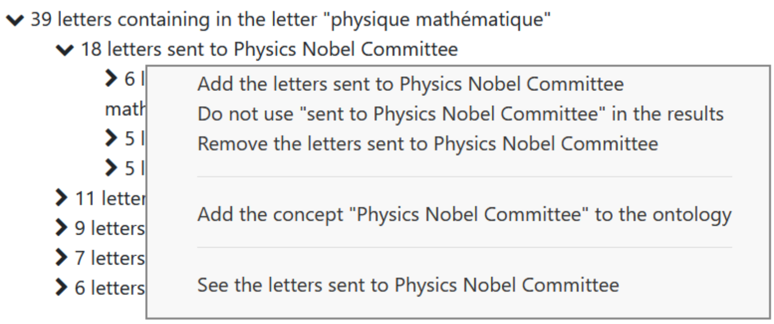


Figure 4: Actions associated to *sent to the Physics Nobel Committee*.

These actions are now detailed:

- *Add the letters sent to the Physics Nobel Committee*: This functionality aims to retrieve additional relevant letters. For example, the user might want to explore the letters sent to the *Physics Nobel Committee* to investigate their connection to mathematics for physics. Triggering this action adds these letters to the analysis set, similar to entering *Physics Nobel Committee* into the recipient field and clicking *Complete the current search*.
- *Do not use "sent to the Physics Nobel Committee" in the result*: This functionality excludes certain properties of FCA grouping to refine hierarchical results by removing irrelevant letter groups. For example, in the running case, it is applied to the *sent to Physics Nobel Committee* group, which is

relevant to the research, but could also be used to exclude irrelevant groups such as *sent by* or *sent to* Henri Poincaré.

- *Remove letters sent to the Physics Nobel Committee*: This functionality eliminates letters not interesting at all for the research problem. For the running example, removing the letters sent to this institute is not adequate but sometimes letters which are not helpful for a given research problem appear and this is this functionality that allows removing them from the analysis.
- *Add the concept "Physics Nobel Committee" to the ontology*: This functionality helps the user to acquire knowledge of the content of the letter. The term *Physics Nobel Committee* is added to the set of unlabeled concepts of the ontology management interface, waiting to be placed in the ontology hierarchy, in this case under the concept *institut* (see Fig. 5).

The next section details the ontology management and its impact on the results.

5. Ontology management

The ontology interface allows for the manual creation of an ontology. It currently manages only atomic concepts and their relations. As shown in Fig. 5, it has two main parts: a *Set of unclassified concepts*, containing candidate concepts added by the user or retrieved from the *Add to the ontology* functionality, and a *Set of classified concepts*, organized hierarchically. For instance, *Physics Nobel Committee* is in the unclassified set, while the classified set includes *mathématiques* with 4 children in the hierarchy, *physique* with 3 children, and *institut* with 3 children. The hierarchy reflects specific/generic relationships between letters. This means that if *fonction* is a child of *mathématiques* in the hierarchy, it has to be interpreted as "a letter annotated by *fonction* will also be annotated by *mathématiques*." Formally, using description logic notation, this means that the following formula is added to the ontology:

$$\texttt{Letter} \sqcap \exists \texttt{isAnnotatedBy}.\{\texttt{"fonction"}\} \sqsubseteq \texttt{Letter} \sqcap \exists \texttt{isAnnotatedBy}.\{\texttt{"mathématiques"}\}$$

Hierarchy management, including moving concepts, is done by drag-and-drop. Here *Physics Nobel Committee* can be moved under *institut* (in English *institute*).

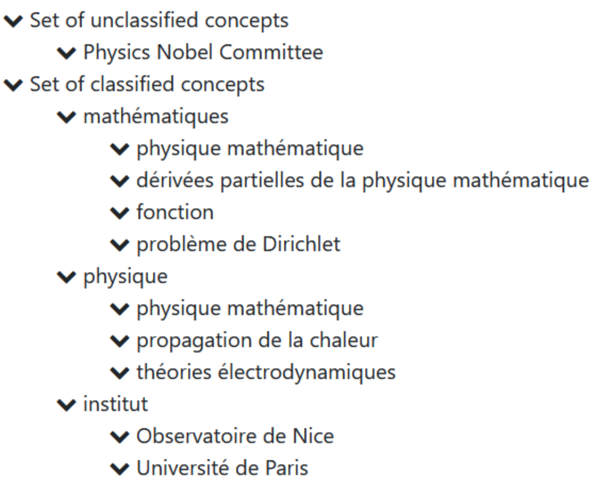


Figure 5: Concept hierarchy management interface. In this hierarchy if C is a child of D then the letters annotated by C are also annotated by D.

This hierarchy is used to improve the FCA process by enriching the binary context with new letter properties. For example, if letter $\texttt{l}_{12}$ has the property *fonction* then it has also the property *mathématiques*. Thus, the historian obtains a more structured result than without this hierarchy. For example, letters containing *propagation de la chaleur* or *théories électrodynamiques* are linked to *physique*, creating a group of letters about physics (see Fig. 6). Similarly, a group of 40 letters related to both *mathématiques* and *physique* is formed, even if they do not contain these exact terms but include related concepts. Furthermore, using the ontology in the search process ensures that entering *physique* retrieves letters associated with the *physique* part of the ontology.

- 63 letters of the current search
  - 42 letters about "physique"
    - 40 letters about "mathématiques"
      - 39 letters containing in the letter "physique mathématique"
        - 18 letters sent to Physics Nobel Committee
        - 7 letters containing in the letter "fonction"
        - 6 letters containing in the letter "problème de Dirichlet"
        - 5 letters containing in the letter "Comité Nobel de Physique"
      - 19 letters sent to Physics Nobel Committee
      - 8 letters containing in the letter "fonction"
      - 6 letters containing in the letter "Comité Nobel de Physique"
      - 5 letters containing in the letter "théories électrodynamiques"
      - ...

Figure 6: The results after having taken into account the concept hierarchy of Figure 5.

6. Discussion and conclusion

This paper presented Harold, an iterative and interactive system for exploring the Henri Poincaré correspondence without requiring knowledge of semantic web technologies. The system enables analysis of both metadata and full text, using FCA to group letters by shared properties. Ontology management, integrated with the exploration system, enhances knowledge discovery by organizing properties hierarchically, which improves both letter organization and retrieval.

The principles used to explore the Henri Poincaré correspondence are applicable to other corpora or data types. An ongoing application of Harold focuses on analyzing computer science publications, particularly in areas such as semantic web, ontology building, and digital humanities, with metadata including authors, publication venue, year, keywords, and for textual data, titles, and abstracts. Future work includes evaluating the system by comparing the results of two historians working on the same research problem, one using Harold and the other not, focusing on coverage, letter retrieval, and new knowledge acquired. Another focus is improving ontology management, including (1) incorporating various relationships between concepts, (2) refining the handling of letter properties (e.g., `exchange with` being more general than `sent to`), and (3) managing concept definitions (e.g., defining Henri Poincaré's study years as between 1873 and 1879).

As Harold is still under development, a proper evaluation has to be performed. It would be necessary to compare how two Henri Poincaré specialists conduct research on a given topic with or without Harold. It should be done in a limited time frame and with the same research question for both historians. The set of letters selected by each researcher would then be analyzed, as only the historian using Harold would have built an ontology. Feedback from these researchers on Harold would also be insightful. Providing public access to an instance of Harold using the Henri Poincaré correspondence or other corpora could allow more people to test the system in the future.

References

[1] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, Berlin, 1999.

[2] E. Kaufmann, A. Bernstein, Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases, Journal of Web Semantics 8 (2010) 377–393.

[3] D. Diefenbach, V. Lopez, K. Singh, P. Maret, Core techniques of question answering systems over knowledge bases: a survey, Knowledge and Information Systems 55 (2018) 529–569.

[4] C. Pradel, O. Haemmerlé, N. Hernandez, Swip: A natural language to SPARQL interface implemented with SPARQL, in: N. Hernandez, R. Jäschke, M. Croitoru (Eds.), Graph-Based Representation and Reasoning, Springer International Publishing, Cham, 2014, pp. 260–274.

[5] G. M. Biancofiore, Y. Deldjoo, T. D. Noia, E. Di Sciascio, F. Narducci, Interactive question answering systems: Literature review, ACM Comput. Surv. 56 (2024) 239:1–239:38. doi:10.1145/3657631.

[6] M. Zaib, W. E. Zhang, Q. Z. Sheng, A. Mahmood, Y. Zhang, Conversational question answering: a survey, Knowledge and Information Systems 64 (2022) 3151–3195. doi:10.1007/s10115-022-01744-y.

[7] Y. Lan, G. He, J. Jiang, J. Jiang, W. X. Zhao, J.-R. Wen, Complex knowledge base question answering: A survey, 2022. doi:10.48550/arXiv.2108.06688.

[8] E. Hyvönen, S. Ferré, Sparklis: An expressive query builder for SPARQL endpoints with guidance in natural language, Semantic Web 8 (2016) 405–418.

[9] A. Russell, P. Smart, NITELIGHT: A Graphical Editor for SPARQL Queries, in: 7th International Semantic Web Conference (ISWC 2008), Poster, 2008.

[10] N. Lasolle, A Navigation Tool for Exploring Semantic Web Corpora, in: Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks, Virtual conference, France, 2021.

[11] O. Bruneau, N. Lasolle, J. Lieber, E. Nauer, S. Pavlova, L. Rollet, Applying and developing semantic web technologies for exploiting a corpus in history of science: The case study of the Henri Poincaré correspondence, Semantic Web 12 (2021) 359–378.

[12] E. Nauer, Y. Toussaint, CreChainDo: an iterative and interactive web information retrieval system based on lattices, International Journal of General Systems 38 (2009) 363–378.

[13] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, H. M. Abbasi, A survey of ontology learning techniques and applications, Database: The Journal of Biological Databases and Curation 2018 (2018).

[14] P. Buitelaar, P. Cimiano, B. Magnini, Ontology Learning from Text: An Overview, volume 123, IOS Press, Amsterdam, 2005, pp. 3–12.

[15] M. D'aquin, TaBIIC: Taxonomy Building through Iterative and Interactive Clustering, in: Formal Ontology in Information Systems, IOS Press, 2023, pp. 155–168.

[16] P. Nabonnand, La Correspondance entre Henri Poincaré et Gösta Mittag-Leffler, Publication des Archives Henri Poincaré, Birkhäuser Basel, 1998.

[17] J.-M. Meunier, S. Szoniecky, D. Berthereau, Utilisation d'Omeka-S pour la conception et le partage de ressources pédagogiques, in: Zotero & Omeka-des outils pour les humanités numériques, Poitiers, France, 2019.

[18] L. Szathmary, A. Napoli, CORON: A framework for levelwise itemset mining algorithms, Supplementary Proceedings of The Third International Conference on Formal Concept Analysis (ICFCA '05), Lens, France (2005) 110–113.