Indexing and Exploring a Digital Humanities Corpus

Nicolas Lasolle

Indexing and Exploring a Digital Humanities Corpus

Nicolas Lasolle

Université de Lorraine, CNRS, Université de Strasbourg, AHP-PReST, F-54000 Nancy, France
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
nicolas.lasolle@univ-lorraine.fr

Abstract. Semantic Web technologies have been chosen to structure and publish data of the Henri Poincaré correspondence corpus and to provide a set of tools for exploiting it. Two major issues have arisen during this work. The first one is related to the manual editing of triples in order to populate an RDF database. It is a tedious task for users with an important risk of error. That justifies the development of a dedicated tool to assist them during this process. It uses and combines different methods which are presented in this article. The second issue is related to the exploration of the corpus database. The need of an approximate and explained search has emerged. An engine based on the application of SPARQL query transformation rules has been designed. Different research tracks are considered to pursue this work which intends to apply in other contexts than the Henri Poincaré correspondence corpus.

Keywords: case-based reasoning, Semantic Web, content annotation, explained and approximate search, RDF(S), digital humanities

1 Introduction

The Henri Poincaré correspondence corpus is composed of around 2100 letters and gathers scientific, administrative and private exchanges. These letters are available on the website `http://henripoincare.fr` which was created and is managed with the CMS[1] Omeka S [1]. In addition to the Omeka S environment, Semantic Web technologies have been applied to this corpus.

Resource Description Framework (RDF [4]) allows the representation of data by using a labeled directed graph. It is based on the use of triples of the form ⟨*subject predicate object*⟩. For instance, ⟨`letter11 sentBy henriPoincaré`⟩ states that letter 11 has been sent by Henri Poincaré. RDFS Schema is extending the RDF model by introducing a new set of classes and properties. It allows to create a hierarchy between classes (using `rdfs:subclassof`) and properties (using `rdfs:subpropertyof`). `rdfs:domain` (resp. `rdfs:range`) applies for a property and adds a constraint about the type of the resource which is in

[1] Content Management System

subject (resp. *object*) position for a triple. Different inference rules which use these properties are considered [2]. SPARQL is a query language for RDF data which is a W3C[2] recommendation [5]. For the sake of readability, its specific syntax is not presented here. In the remainder of this article, queries are presented in an informal way. Manual Semantic Web data editing (i.e. creating the triples) is often a tedious task which may cause errors of different kinds. A suggestion tool combining the use of Semantic Web technologies with a case-based reasoning (CBR [6]) methodology is explained in Section 2. The use of SPARQL querying is sometimes not satisfactory to exploit this corpus. A tool associated with a language, named SQTRL (SPARQL Query Transformation Rule Language) have been designed to enable flexible querying [3]. This mechanism as well as different future works are presented in Section 3. Section 4 concludes.

2 Assisting human annotation for Semantic Web data editing thanks to a CBR methodology

2.1 Proposal of an annotation tool

A tool including an autocomplete mechanism has been created to assist users during the annotation process. It enables the visualization and the update of RDF databases. An *annotation question* corresponds to a triple for which 1,2 or the 3 fields are unknown, and for which a field is currently being edited. For example, consider the annotation question ⟨`letter11 sentTo` `?o`⟩. The objective here is to provide appropriate suggestions by listing and ranking the potential values for the object field. 4 versions of the editor have been implemented and use different approaches to order the suggestion list.

The *basic editor* uses the alphabetical order to rank the potential values.

The *deductive editor* benefits from the knowledge about the `rdfs:domain` and `rdfs:range` of the properties defined within the ontology. For the running example, as `Person` is `range` of the property `sentTo`,[3] this knowledge is used to favor the instances of this class.

The *case-based editor* follows the case-based reasoning methodology(CBR [6]). Information from situations similar to the current annotation question is reused. An annotation problem $\mathtt{x}^{\mathtt{tgt}}$ is composed of an annotation question and a *context* which corresponds to the set of edited triples related to the resource currently being edited. The annotation problem is defined as follows:

$$\mathtt{x}^{\mathtt{tgt}} = \left| \begin{array}{ll} \textbf{question:} & \langle \mathtt{letter11\ sentTo\ ?o} \rangle \\ \textbf{context:} & \langle \mathtt{letter11\ sentBy\ henriPoincaré} \rangle \\ & \langle \mathtt{letter11\ hasTopic\ écolePolytechnique} \rangle \\ & \langle \mathtt{letter11\ quotes\ paulAppell} \rangle \end{array} \right|$$

A solution $\mathtt{y}^{\mathtt{tgt}}$ corresponds to a value for `?o`. The RDF database $\mathcal{D}$ is defined as the case base. Each resource of $\mathcal{D}$ is defined as a source case $\mathtt{x}^{\mathtt{s}}$ and is used

[2] World Wide Web Consortium

[3] According to the Henri Poincaré corpus ontology.

to propose a candidate solution $\mathtt{y}^s$. The method consists in retrieving the most similar resources in the case base. SPARQL querying is used in that context. An initial query `Q` is created to retrieve the resources whose context is the same as the one of `letter11`. However, it is uncommon to find two resources with the exact same context. The SQTRL tool [3], whose functioning is detailed in the following section, is reused in this application framework. Using this tool allows the generation of a new set of queries which can be executed to find resources that partially match the context of `letter11`.

The *combination editor* combines the use of RDFS deduction with CBR.

2.2 Evaluation and results

Two different evaluations, which include a comparison of the different versions of the suggestion system, have been carried out. The RDF graph of the Henri Poincaré correspondence corpus $\mathcal{G}_{\mathrm{HP}}$ has been used as a test set.

The first evaluation is human-based: a user works with the dedicated tool to edit a set of unpublished letters. The user feedback has been collected through a survey in which he has been asked to attribute a score to measure the efficiency of the different versions. The second evaluation is automatic through a program which computes measures to compare the different versions for similar annotation questions. Several edited triples have been randomly extracted from this database graph and have been used to simulate annotation questions for which the answers are already known. The results show that the combination editor is the most efficient, and this for all the properties of the evaluation. Combining the use of RDFS knowledge with a CBR methodology is an appropriate solution to rank the potential values in a efficient way.

3 Approximate and explained search

3.1 SQTRL

SQTRL is a tool which has been designed to manage approximate search and has proven useful in different contexts including the search in the Henri Poincaré correspondence corpus and the case-based cooking system Taaable [3]. Users can configure SPARQL query transformation rules which can be general (e.g. property and class generalization, triple removal, etc.) or context-dependent (e.g. exchange of the sender and the recipient of a letter, substitution of a person by one of his/her colleagues, etc.). An application of a rule r to a SPARQL query $\mathtt{Q}$ for a given RDF database $\mathcal{D}$ could generate a new query $\mathtt{Q}_N$. The execution of $\mathtt{Q}_N$ on $\mathcal{D}$ may return a different set of results than the execution of $\mathtt{Q}$ on that same database. A search tree can be explored, starting from the initial query $\mathtt{Q}$, by applying one or several successive transformation rules. To each rule is associated a cost (defined as a constant integer) which corresponds to a query transformation cost. By defining a maximum cost, it is possible to limit the

depth of the search tree exploration. Let $\mathcal{Q}$ be an example of a query formulated by a historian to query the database of the correspondence corpus $\mathcal{D}_{\text{HP}}$:

$$\mathcal{Q} = \left| \begin{array}{l} \text{“Give me the letters sent by } \texttt{henriPoincaré} \\ \text{ to a physicist with } \texttt{optics} \\ \text{ as a topic between 1880 and 1890.”} \end{array} \right.$$

The queries $\{\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3\}$ correspond to queries generated at depth 1:

$$\mathcal{Q}_1 = \left| \begin{array}{l} \text{“Give me the letters sent by} \\ \underline{\text{ a physicist to } \texttt{henriPoincaré}} \\ \text{with } \texttt{optics} \text{ as a topic} \\ \text{between 1880 and 1890.”} \end{array} \right. \quad \mathcal{Q}_2 = \left| \begin{array}{l} \text{“Give me the letters sent} \\ \text{ by } \texttt{henriPoincaré} \text{ to a} \\ \underline{\text{scientist}} \text{ with } \texttt{optics} \text{ as a} \\ \text{topic between 1880 and 1890.”} \end{array} \right.$$

$$\mathcal{Q}_3 = \left| \begin{array}{l} \text{“Give me the letters sent by } \texttt{henriPoincaré} \\ \text{ to a physicist with } \texttt{optics} \\ \text{ as a topic } \underline{\text{between 1875 and 1895}}\text{.”} \end{array} \right.$$

$\mathcal{Q}_1$ is generated by applying a rule which exchanges the sender and the recipient of the letter. An object class generalization rule replaces `Physicist` in `Scientist` and generates the query $\mathcal{Q}_2$. $\mathcal{Q}_3$ is generated by applying a rule which extends the temporal bounds of the query. Other rules could be applied to Q. Moreover, depending on the maximum cost which has been set, the tree exploration can be continued to generate other queries.

3.2 Future works and research plan

Works related to this approximate and explained search mechanism are frequently being discussed with historians of science. The objective is to provide tools which would assist them efficiently for the study of the Henri Poincaré corpus and which may be applied in other contexts. At this stage, different research tracks are considered and these discussions should be pursued in order to identify the main concerns and specify the research plan.

A first issue is related to the representation of transformation rules which does not insist on the explainability of this mechanism. Moreover, it could be relevant to include user preferences to favor the application of some rules in given situations. The application of a CBR methodology could be useful to store and reuse successful transformation paths. Another issue is related to the management of the costs associated to transformation rules. These have been set subjectively (defined as constant integers) for a first set of rules. It could be relevant to consider it as dependent on the resources occurring in the SPARQL query on which the rule is applied. The CBR community could be helpful here: the issue is to find how to assess similarity between resources to propose a method adapted for the context of the Henri Poincaré correspondence corpus graph. Other issues are related to the application of these transformation rules. As an example, a limitation is related to the occurrence of some unnecessary compositions of rules.

Consider the example presented in Section 3 in which the query $\mathcal{Q}$ is generated into a query $\mathcal{Q}_1$ by applying a rule which exchanges the sender and recipient of the letter. Currently, nothing is done to prevent the application of the same rule on $\mathcal{Q}_1$. It is thus possible to generate, at depth 2, a query $\mathcal{Q}_{11}$ such as $\mathcal{Q}_{11} = \mathcal{Q}$. A way to prevent such a situation should be explored.

4 Conclusion

Semantic Web technologies have been used to exploit the Henri Poincaré correspondence corpus. In that context, several issues have arisen. A tool has been proposed to assist users during the human annotation process. It benefits from the use of RDFS deduction and from the application of a CBR methodology. A human and an automatic evaluation have been carried out for this system and have shown positive results. Another issue is related to the need of an approximate search mechanism. The SQTRL tool proposes a method based on the use of SPARQL query transformation rules. However, the work related to this mechanism should be pursued in order to propose tools and methods suitable for the Henri Poincaré correspondence corpus and which can be reused in other contexts. The use of a CBR methodology is considered to achieve this goal and respond to some issues which have arisen.

Acknowledgement. This work is supported partly by the french PIA project "Lorraine Université d'Excellence", reference ANR-15-IDEX-04-LUE.

References

1. Boulaire, C., Carabelli, R.: Du digital naive au bricoleur numérique les images et le logiciel Omeka. In: Expérimenter les humanités numériques. Des outils individuels aux projets collectifs. Les Presses de l'Université de Montréal (2017)
2. Brickley, D., Guha, R.V.: RDF Schema 1.1, `https://www.w3.org/TR/rdf-schema/`, W3C recommendation, last consultation: February 2020 (2014)
3. 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. Lecture Notes in Computer Science, Springer (2017) 76–91
4. Lassila, O., Swick, R.R., et al.: Resource description framework (RDF) model and syntax specification. (1998)
5. Prud'hommeaux, E.: SPARQL query language for RDF, W3C recommendation (2008)
6. Riesbeck, C.K., Schank, R.C.: Inside Case-Based Reasoning. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey (1989)