Building semantic metadata for historical archives through an ontology-driven user interface

Building Semantic Metadata for Historical Archives through an Ontology-driven User Interface

Annamaria Goy
Dipartimento di Informatica, Università di Torino, Torino, Italy, annamaria.goy@unito.it

Davide Colla
Dipartimento di Informatica, Università di Torino, Torino, Italy, davide.colla@unito.it

Diego Magro
Dipartimento di Informatica, Università di Torino, Torino, Italy, diego.magro@unito.it

Cristina Accornero
Dipartimento di Studi Storici, Università di Torino, Torino, Italy, accornerocristina@gmail.com

Fabrizio Loreto
Dipartimento di Studi Storici, Università di Torino, Torino, Italy, fabrizio.loreto@unito.it

Daniele Paolo Radicioni
Dipartimento di Informatica, Università di Torino, Torino, Italy, daniele.radicioni@unito.it

ABSTRACT

Historical archives represent an immense wealth, the potential of which is endangered by the lack of effective management and access tools. We believe that this issue can be faced by providing archive catalogs with a **semantic layer**, containing rich semantic metadata, representing the content of documents in a full-fledged formal machine-readable format. In this paper we present the contribution offered in this direction by the **PRiSMHA project**, in which the conceptual vocabulary of the semantic layer is represented by computational ontologies. However, acquiring semantic knowledge represents a well-known bottleneck for knowledge-based systems: in order to solve this problem, PRiSMHA relies on a **crowdsourcing collaborative model**, i.e., an online community of users who collaborate in building semantic representations of the content of archival documents. In this perspective, this paper aims at answering the following **research question**: *Starting from the axioms characterizing concepts in the computational ontology underlying the system, how can we derive a user interface enabling users to formally represent the content of archival documents by exploiting the conceptual vocabulary provided by the ontology?*
Our solution includes the following steps: **(a)** A manually defined configuration, acting as a pre-filter, to hide "unsuited" classes, properties, and relations; **(b)** An algorithm, combining heuristics and reasoning, which extracts from the ontology all and only the "compatible" properties and relations, given an entity (event) type. **(c)** A set of strategies to rank, group, and present the entity (event) properties and relations, based on the results of a study with users. This integrated solution enabled us to design an **ontology-driven user interface** enabling users to characterize entities, and in particular (historical) events, on the basis of the vocabulary provided by the ontology.

CCS CONCEPTS

• Digital libraries and archives • Semantic networks • Ontology engineering

KEYWORDS

Ontology-driven User Interfaces, Computational ontologies, Historical Archives, Crowdsurcing platform

*POSTPRINT VERSION. Cite as: Goy, A.; Colla, D.; Magro, D.; Accornero, C.; Loreto, Radicioni, D.P. Building Semantic Metadata for Historical Archives through an Ontology-driven User Interface. *Journal on Computing and Cultural Heritage*, 13(3), 2020, ISBN 978-989-758-417-6, pages 1-36.

1 Introduction

Historical archives, storing documentary about social and political history of the 20th century, represent an immense wealth, that could be exploited to support historical memory, cultural identity and integration, as well as the understanding of social changes and the ability to manage them. However, this potential is endangered by the lack of effective management and access tools, able to orientate the audience in the extremely rich and heterogeneous universe of documents and resources. In particular, in Italy, beside some interesting projects (e.g., ArCo: dati.beniculturali.it/progetto-arco-architettura-della-conoscenza; Memoranea: www.memoranea.it; Mèmora: www.memora.piemonte.it; among many others), historical archives often contain non-digitized resources (being them paper-based or audio-visual resources stored on out-of-date supports) and very poor, non-standard, and sometimes paper-based, catalogs.
ICT could provide a great help on many fronts:

(A) Document digitization and OCR for textual resources.
(B) Catalog integration, through Service Architectures (Alonso et al., 2014) and data linking, e.g., through the Linked Open Data paradigm (Heath and Bizer, 2011).
(C) Metadata enrichment, by adding semantic information about resource content: many research fields could support this task, like Natural Language Processing -- e.g., (Sleimi et al., 2018; Carducci et al., 2019), image recognition and, in particular, hand-written text recognition -- e.g., (Vamvakas et al., 2010), crowdsourcing models -- e.g., (Terras, 2016).
(D) Effective and user-friendly access tools, based on HCI and UX Design principles -- e.g., (Sharp et al., 2019).

The research activity presented in this paper focuses on point (C): we believe that the mentioned lack of effective management and access tools, a huge problem given the immense documentary heritage available even just in Europe, can only be faced by implementing a hybrid strategy, that integrates automatic techniques and user-generated content (Foley et al., 2017). In this paper we present the contribution offered in this direction by the PRiSMA project (di.unito.it/prismha).
PRiSMHA (Providing Rich Semantic Metadata for Historical Archives) (Goy et al., 2017) is a three-year (2017-2020) Italian project, funded by Compagnia di San Paolo Foundation and Università di Torino. It is carried out by the Computer Science and the Historical Studies Departments of the University of Torino, in collaboration with Polo del ‘900 (www.polodel900.it), a cultural institution co-funded by Comune di Torino, Regione Piemonte, and Compagnia di San Paolo Foundation, including 22 partners. Polo del ‘900 boasts an extremely rich integrated archive -- (partially) accessible through the online platform *9centRo* (www.polodel900.it/9centro) -- 25% of which is represented by the Fondazione Istituto piemontese Antonio Gramsci’s collections (www.gramscitorino.it).
In order to develop a proof-of-concept prototype of our solution for metadata enrichment (see Section 4), we identified the portion of the Istituto Gramsci’s collections containing documents related to the students and workers protest during the years 1968-1969 in Italy (Goy et al., 2019a): the majority of these documents is represented by non-digitized typewritten leaflet with manual annotations and drawings, together with pictures and newspaper articles (Figure 2).
PRiSMHA starts from the assumption that only a layer of semantically rich metadata can support an actual enhancement of the access to archival resources, an approach supported by several researchers in the area of Digital Humanities; see, for instance, (Motta et al., 2000). Exactly as a good human archivist, a smart digital one needs to “know” the content of the available documents in order to be able to retrieve all and only the relevant ones, independently of the words actually used to report them in the primary sources; a case study supporting this claim can be found in (Goy et al., 2019b).
This implies that the system needs a detailed semantic knowledge of the domain, i.e. of topics of the documents: in PRiSMHA, such knowledge is provided by computational ontologies (see Section 3), which represent the system conceptual “vocabulary” (Goy et al., 2015) for expressing the semantically rich metadata layer. This layer consists in a knowledge base containing a formal, machine-readable full-fledged description of the content of archival documents.

However, acquiring such knowledge represents a well-known bottleneck for knowledge-based systems, that can threaten the feasibility of the approach (Foley et al., 2017). In order to provide a contribution to the solution of this problem, PRiSMHA relies on a crowdsourcing collaborative model (Terras, 2016), (Noordegraaf et al., 2014), i.e., an online community of users -- including historians, archivists, students, or simply enthusiasts -- who collaborate in building semantic representations of the content of archival documents. Moreover, when full text (e.g., OCR-ized documents) is available, various sorts of processing can be employed -- aimed at supporting the annotation process with helpful suggestions -- such as automatic Information Extraction techniques (Boschetti et al., 2014), Named Entity Recognition (Nadeau and Sekine, 2007), Event Mining (Hogenboom et al, 2011), and Semantic Role Labeling (Palmer et al., 2010). Such topics have been left out of the scope of this paper, that focuses on demonstrating the feasibility of an ontology-driven web-based platform enabling users to build rich semantic metadata over archival resources. In particular, the main challenge of the research reported in this paper is the exploitation, within the crowdsourcing platform, of the formal conceptual vocabulary provided by the computational ontology.
As stated above, the ontology represents the system conceptual vocabulary, which -- in order to be rich enough to express document content -- is necessarily complex. Therefore, the main challenge of a user interface based on such ontology is to exploit this semantic complexity by offering users, at the same time, a simple and friendly way to "write" semantic meta-data.
In this framework, PRiSMHA complies with the approach, widely shared in the research community studying the use of ICT and Semantic Web to handle historical documents (see Section 2), centered on the notion of *event* and on its characterization in terms of its properties (more specifically, the time and place of the event, and the involved participants) and relations with other events (e.g., the notion of *cause*, *influence*, etc.). The characterization of event typologies, properties, and relations, will be described in Section 3, while the ontology-driven user interface will be presented in Section 4. We start by formulating the main **research question** of the work presented in this paper:

Starting from the axioms characterizing events and their properties/relations in the computational ontology underlying the system, how can we derive a user interface enabling users to characterize events by exploiting the conceptual vocabulary provided by the ontology itself?

First of all, we have applied a simple (handcrafted) filter that excludes from the user interface classes, properties, and relations considered "unsuited" for the users interacting with the crowdsourcing platform: the tool enabling administrators to configure class/property/relation visibility is described in Section 4.
However, the main issue of an ontology-driven user interface is to single out, from the axioms of the ontology, all and only the properties and relations that are "compatible" with a given event type (Gonçalves et al. 2017), i.e. -- in RDF terms (see Section 4) -- that can be asserted in a triple having an instance of that event type as subject. This extraction is not trivial: for instance, according to the ontology, does the property *hasAgent* make sense to describe an event of type *StreetClash*? And to describe a *PersonBirth*? and an *Earthquake*? And what about the relation *hasPurpose* to describe a *Marriage* or a *Strike*? The algorithm used to perform such an extraction is described in Section 5.1. Nonetheless, the available properties selected by the algorithm result in a quite large set, possibly undermining the usability and friendliness of the user interface (see Figure 5). We thus designed a user study to ask for users help in defining strategies to rank properties and relations on the basis of their relevance (for describing a given event type). The resulting ranking would enable us to group properties and relations to build a better user interface (see Sections 5.2 and 5.3).
In summary, the main contribution of this paper is a solution for building a user-friendly interface enabling users to characterize events on the basis of the (complex) conceptual vocabulary provided by a rich computational ontology, and overcoming the mentioned problems. As already stated, our solution includes the following steps:

- A manually defined configuration, acting as a pre-filter (described in Section 4).
- An algorithm, combining heuristics and reasoning, which extracts from the ontology all and only the "compatible" properties and relations, given an event type (described in Section 5.1).

- A set of strategies to rank, group, and present the event properties and relations, based on the results of a study with users (described in Section 5.2).

The resulting user interface is described in Section 5.3, while Section 6 concludes the paper by sketching the future directions of our work.

2 Related Work

The first research area to be considered is represented by the work about ontology-driven User Interfaces (UI). A good starting point on this topic is the survey by Paulheim and Probst (Paulheim and Probst, 2010), where the authors offer a detailed analysis of *ontology-enhanced* UI, focusing on the different purposes for using ontologies in UI, from which different requirements are derived. The definition proposed by Paulheim and Probst is the following: "An *ontology-enhanced user interface* is a user interface whose visualization capabilities, interaction possibilities, or development process are enabled or (at least) improved by the employment of one or more ontologies" (Paulheim and Probst 2010, p. 37). The authors prefer "the more general notion *ontology-enhanced user interface* instead of *ontology-driven user interface* [...] since ontologies may also be employed to provide one single functionality in a larger user interface (and thus *enhance* the user interface) without being the key element *driving* the user interface" (Paulheim and Probst 2010, p. 37). With respect to this perspective, in our approach the UI is actually ontology-*driven* (and not only *enhanced*). The analysis of systems having an ontology-enhanced UI takes into account different parameters, that can be used to classify them: Table 1 shows the parameters and the possible values for each of them (according to Paulheim and Probst), together with the choices made in the PRiSMHA project.

Table 1: The parameters to analyze ontology-enhanced UI proposed by Paulheim and Probst (Paulheim and Probst, 2010).

Parameter (criterion)	Possible values	PRiSMHA
Domain	real world, IT system, users and roles	real world
Complexity	informal, low, medium, high	high
Time	design time, run time	run time
Visualization	no presentation, lists, graphical, verbalized	mixed
Interaction	no interaction, view only, view and edit	view only
Storage	central, distributed	central
Grounding in foundational ontologies	free floating, grounded	grounded
Modularity	one module, several modules	several modules

Moreover, such classification parameters are intersected with the purpose for which ontologies are used: ontologies at the UI level can be used to improve (a) the visualization capabilities, (b) the interaction possibilities, (c) the development process. In PRiSMHA, the ontology is used for improving the interaction possibilities (case (b) above). As far as this goal is concerned, Paulheim and Probst analyze different

approaches, namely *ontology-based browsing*, *user input assistance*, *providing help*, *user interface integration* (Paulheim and Probst, 2010). In PRiSMHA, as stated above, the ontology drives the user interaction and thus it is used to provide user with input assistance. In the authors words: “In a form-based user interface, not every combination of input options is feasible. An ontology formalizing knowledge about the domain of the objects whose data is entered in a user interface may be employed to provide plausibility checking on the user's input.” (Paulheim and Probst 2010, p. 49). Moreover, the authors state that “plausibility checking can help reducing the complexity of input forms. Thus, plausibility checking can improve the usability of a user interface.” (Paulheim and Probst 2010, p. 49). The authors add that textual input can be improved by autocompletion supported by terms defined in the ontology. As we will see in Sections 4 and 5, ontology-supported plausibility checking, form menus, and autocompletion for textual input are precisely the tools adopted in PRiSMHA to generate the ontology-driven UI: we thus share with the authors the claim that this approach “ensures that the user only enters terms that the system understands and follows the idea of using ontologies as a shared vocabulary (Gruber, 1995)” (Paulheim and Probst 2010, p. 50).

A large part of the literature about ontology-based UI concerns *Ontology-Based Data Access*; among many others, see (Calvanese et al., 2015) for an example in the historical domain, and (Soylu et al., 2017) that, besides proposing an ontology-based query formulation system, also contains a survey of such systems, including *faceted search* (Tunkelang and Marchionini, 2009) and *query by navigation* (see, for example, (Franconi et al., 2010), where ontology navigation is coupled with Natural Language Generation techniques).

As far as the work presented in this paper is concerned, we are interested in the use of ontologies in *data acquisition* systems: from this point of view, ontologies are often employed to generate structured *web forms*; see, for instance, (Giretzlehner et al., 2012; Horridge et al., 2014). However, as clearly stated by Gonçalves and colleagues (Gonçalves et al. 2017), “The rise of the Web Ontology Language (OWL) [...], standardized by the World Wide Web Consortium (W3C) in 2004, caused a paradigm shift in knowledge representation from frame-based to axiom-based. Because of its axiom-based nature, it is more difficult to acquire instance data based on OWL than it was based on frames. With OWL as the preferred modeling language for ontologies, class definitions are collections of description logic (DL) axioms, and can no longer be seen as templates for forms (Rector 2013)” (Gonçalves et al. 2017, p 1). In particular, Gonçalves and colleagues designed a system in which the form layout is itself specified by an ontology. In contrast, in PRiSMHA, the ontology only defines the possible *content* of the forms, i.e. it provides users with the available values to be selected (see Section 4).

Another research area that is relevant for PRiSMHA, although the issues it covers fall partially outside the main focus of this paper, is represented by semantic models used in Digital Humanities. In at least the last decade, computational ontologies, Linked Data, and in general approaches based on semantic models have been largely used to enhance access and management of Cultural Heritage datasets (Meroño-Peñuela et al., 2015; Oomen and Belice, 2012; Calvanese et al., 2015). Many initiatives demonstrate the interest for Semantic Web tools in the Digital Humanities: The huge European Union digital platform Europeana (www.europeana.eu) and projects like WarSampo (seco.cs.aalto.fi/projects/sotasampo/en) or PAPYRUS (Katifori et al., 2011) -- using CIDOC Conceptual Reference Model (www.cidoc-crm.org) and Linked Open Data -- among many others.

Semantic models underlying Digital Humanities projects are often driven by the notion of *event* and its participants, in particular in the historical domain (Sprugnoli and Tonelli, 2017; Nanni et al., 2017; Zarri, 2015). However, such semantic models, if compared with our approach, usually have a lighter axiomatization, which means a loose formal characterization of the modeled concepts, properties, and relations. Specifically, they usually include (if any) a weak characterization of domain-specific event typologies, properties, relations between events (e.g., *cause*), and relations between events and entities (e.g., *participation*) (Goy et al., 2018). For example, the semantic model underlying the Europeana initiative, EDM (Europeana Data Model) (Isaac, 2013; Europeana, 2016) offers the concepts of event with its basic properties (i.e., participants, time, and location), but does not provide any further characterization. The Event Ontology (purl.org/NET/c4dm/event.owl) (Raymond and Abdallah, 2007), only represents events with their basic features and models a few participation modalities, while LODE (Shaw et al., 2009), the result of the alignment of different existing ontologies (including Event Ontology and CIDOC-CRM), supports the

representation of events with participants, time and location, but, again, no role and no specific event typology are available. Projects like Agora (van den Akker et al., 2010) and DIVE (de Boer et al., 2015) are based on SEM (Simple Event Model) (van Hage et al., 2011) a domain-independent lightweight ontology including concepts such as *event, participant* (called *actor*), *time, space*, and *role* with no further specification of participation modalities or event types. The same remarks hold for the ontology used in the CultureSampo project (Hyvönen et al., 2012). CIDOC-CRM (Doerr, 2003; Le Beuf et al., 2015) offers about thirty rather general event types, along with dozens of properties suitable for describing events. However, as the above-mentioned ones, also the CIDOC-CRM ontology is loosely axiomatized and quite general. Finally, the Event Model F (Scherp et al., 2009) is a top-level ontology extending DnS (Gangemi and Mika, 2003), and thus it inherits the DnS semantically rich characterization of events, participation, and roles played by participants, although it is not tailored to the historical domain.

In some application contexts, such ontologies might not be enough, because either a stronger formalization is needed or a more specific characterization of the domain concepts is required. However, also in those contexts, these ontologies can still play an important role, precisely by virtue of their generality and, at least for some of them, of their popularity. They can actually be targets of ontology alignments aimed at mapping more specific or more formalized concepts to their ones: In this way, they represent a general common layer, which enhances data interoperability. As an example of this approach, it is worth mentioning the Data for History initiative (dataforhistory.org), which offers a web-based platform supporting an open ontology development process and the alignment of the developed ontologies to CIDOC-CRM.

From the system architecture point of view, an interesting perspective to be taken into account is proposed by Dragoni and colleagues (Dragoni et al., 2016). The authors describe a general architecture (and a tool implementing it, called MOKI-CH) for knowledge management platforms in the Cultural Heritage domain. The authors identify several requirements for such an architecture, namely: (a) collaboration support in building semantic models of the domain, (b) data exposure and data linking, (c) multilingual management, (d) user engagement. With respect to requirement (a), this paper faces the issues raising from the use of computational ontologies of the domain as a semantic model driving the collaborative platform. Differently from the corresponding requirement in (Dragoni et al., 2016), however, the collaboration in PRiSMHA aims at building a knowledge base of semantic metadata, and not the ontological model itself. As far as data exposure and data linking is concerned (requirement (b)), we are working at exposing a SPARQL endpoint, making PRiSMHA knowledge base available for third party applications (see Section 4). Multilinguality (requirement (c)) is an important issue to be handled, especially in a European perspective, and it is part of our future work, while user engagement (requirement (d)) definitely deserves a deeper study: motivating users to participate in crowdsourcing projects is of paramount importance in order to build solid and effective communities, and it requires the close collaboration with local institutions, that in the case of PRiSMHA, are represented by the cultural organizations involved in the Polo del ‘900.

Public engagement through ICT in the field of Cultural Heritage has been largely promoted and discussed in the last decades (Visser and Richardson 2013; King, et al., 2016). In particular, crowdsourcing approaches have been adopted in order to perform different kinds of tasks -- ranging from handwritten documents transcription to old picture recognition, segmentation, and classification (Terras, 2016) -- and several cultural institutions have started crowdsourcing projects (Ashenfelder, 2015). A good survey, analyzing success factors of crowdsourcing projects, can be found in (Noordegraaf et al., 2014). Moreover, two projects, among many others, are worth mentioning: Scribe (www.nypl.org/blog/2015/11/23/scribe-framework-community-transcription) is an open-source platform enabling communities of users to transcribe documents like handwritten texts, that cannot be successfully processed by OCR tools; Micropasts (crowdsourced.micropasts.org) (Bonacchi et al 2014; Bonacchi et al 2019) supports the collaborative production of meta-data for historical documents (the location of scenes appearing in pictures, the transcription of letters, etc.). One of the most interesting exploitation of crowdsourcing in the Cultural Heritage domain is represented by user-generated meta-data, usually consisting in social tagging, which typically leads to *folksonomies* (Hooland et al., 2011; Ridge, 2013; Gupta et al., 2010). Social tagging involves important issues such as the quality of the produced meta-data, the personal and social acceptance of participatory platforms, and the already mentioned motivations of participants (Koukopoulos and

Koukopoulos, 2019), but the discussion of these aspects falls outside the scope of this paper. Another close perspective that is worth mentioning is that of groups of experts collaborating in the annotations of heritage collections, as, for example, in the CULTURA project (Agosti el al., 2013) or in the SAGE environment proposed by Foley and colleagues (Foley et al., 2017), where user-generated annotations are supported by automatic techniques. Moreover, the goal of Foley and colleagues is to build semantically rich metadata, in the sense discussed in (Marchetti et al., 2007). Both the integration of automatic Information Extraction tools to support annotators and the goal of producing rich (full-fledged) semantic metadata are goals of the PRiSMHA project, too.

3 The Ontology: HERO

HERO (Historical Event Representation Ontology) is a modular ontology that covers the different aspects of historical events, both at general and at domain-specific levels. As already mentioned in Section 2, we can consider the backbone of a typical historical event description as the specification of the following basic elements: The event type (e.g., a murder), the place where it occurred (e.g., a specific building in Rome), the time when it took place (e.g., in September 1972), and the participants in the event (e.g., those three people, that revolver, etc.).[1] The organization of the HERO ontology mirrors such general structure of an event description.

An upper-level module, called HERO-TOP (https://w3id.org/hero/HERO-TOP), provides the most general notions that are inherited by all the other modules. In particular, following the ontological principles underpinning the DOLCE ontology (Borgo and Masolo, 2009), this module accounts for the basic distinctions between *perdurants*, *objects*, and *abstract entities*.

Perdurants happen in time and evolve by accumulating temporal parts; examples of perdurants are a person birth, an assassination, a war, etc. *Objects* are those entities that *are in time* and that are "wholly present at any time they are present" (Masolo et al., 2003, p. 15); there can be *physical objects* (e.g., mountains, houses, persons, etc.) and *non-physical objects*. Among non-physical objects, *social objects* (i.e., objects depending on a community of agents) play a major role in the historical domain (examples of social objects are laws, social roles, organizations, etc.). The basic relationship between objects and perdurants is *participation*, i.e. objects participate in perdurants. For instance, king Umberto I, Gaetano Bresci, and the revolver used by the latter to shoot the former are three objects participating in the assassination of Umberto I by Gaetano Bresci (a perdurant). Abstract entities are neither objects nor perdurants. The most important class of abstract entities in HERO is that of *time intervals*, intended as in classical physics (such as July 29 in the year 1900, April 1945, the year 1968, etc.).

The HERO-EVENT module (https://w3id.org/hero/HERO-EVENT) accounts for the basic notions related to event representation. In particular, it distinguishes between two kinds of perdurants, i.e. *events* (in strict sense), which represent changes in some state of affairs (e.g., the assassination of king Umberto I) and *states* (which represent the persistence of some state of affairs, such as the Italian unemployment rate being 11.1% on September 2017). Among events, we can distinguish *actions* (i.e., intentional events, such as the assassination of Umberto I) from *phenomena* (i.e., non-intentional events, such as Kobe earthquake in 1995). HERO-EVENT also characterizes some common event types, such as: *coming into existence*/*ceasing to exist* (e.g., the Italian Communist Party came into existence on January 21st, 1921 in Livorno, and ceased to exist in February 1991 in Rimini), *person birth*/*person death*, *entity creation* (e.g., on January 21st 1921 in Livorno Antonio Gramsci and other people founded the Italian Communist Party) and *entity destruction* (e.g., on February 1991 in Rimini the delegates to the 20th Congress approved the dissolution of the Italian Communist Party), *role assumption*/*ceasing playing a role* (e.g., Barack Obama became president of the United States in 2009 and ceased playing that role in 2017), etc.

[1] It is worth noting that we should not consider such elements as mandatory in any event description. In some cases, some of them may be unknown or irrelevant. By contrast, in some cases other kinds of elements should be specified, possibly in addition to those listed above (e.g., the prospective participants -- i.e., people having the right to vote -- in a voting event).

HERO-EVENT puts at disposal several properties for describing events and states. Some of them allow us to specify *where* an event or a state occurred, as well as the *initial*, *final* and *intermediate locations* for those events or states that change their place during their evolution (e.g., a protest march). Other properties allow us to specify *when* an event or a state took place, i.e. its (possibly approximate) *beginning*, *ending times*, and *timespans*. Another set of properties of paramount importance are those representing the so-called *thematic* (or *semantic*) *roles*. Such properties express the participation modalities in events or states, i.e. the roles that participants can play in them (Goy et al., 2018) (e.g., agent, patient, instrument, opponent, etc.). For instance, king Umberto I participated in his assassination as a *patient* (since he was affected by the event), while Gaetano Bresci participated as an *agent* (since he intentionally carried out the event) and the revolver participated as an *instrument*. Some properties represent relations between events/states: Among them, we find a relation for expressing the fact that an event/state somehow *influenced* another event/state (of which *causality* is a specific case) and that an event (state) is a *sub-event* (*sub-state*) of another event (state).

The HERO-PLACE module (https://w3id.org/hero/HERO-PLACE) characterizes the basic notions relevant to places (intended in a broad sense as geographic features, i.e., intuitively, as anything that we can represent on a map). In particular, HERO-PLACE distinguishes between *natural places* (e.g., mountains, seas, etc.) and *artificial places* (e.g., buildings, dams, etc.); between *terrestrial places* (e.g., terrestrial mountains, rivers, etc.) and *astronomical places* (e.g., planets different from the Earth, stars, etc.). Moreover, it offers a set of properties for representing mutual relations between places (e.g., inclusion, overlapping, etc.).

The HERO-TIME module (https://w3id.org/hero/HERO-TIME) accounts for the basic notions relevant to time (in its classical understanding and, in particular, related to the description of human affairs). It enables us to represent time intervals and durations, and to reason on them, also by means of Allen's relations (Allen, 1983). For those time intervals that can be specified by referring to clock/calendar conventions (e.g., days, months, years, etc.), the corresponding expressions (e.g., dates, months and year expressions, etc.) can be represented as well. Given their relevance in the historical domain (as in many others), HERO-TIME provides also the basics for representing temporary relationships (e.g., temporary parthood, location, affiliation in organizations, etc.).

The HERO-ROCS module (https://w3id.org/hero/HERO-ROCS) characterizes some notions of great importance in the historical domain. It accounts for (social) roles (such as professions, public offices, etc.), organizations (such as companies, public bodies, etc.) and collective entities (e.g., laborers, consumers, groups of citizens, etc.), i.e., "collections of individual objects (their members) that have their own individuality, full existence, and the capability to take part in (historical) events; they can be ascribed characteristics, properties or behavior that cannot be (conveniently) reduced to those of their members" (Goy and Magro, 2019).

HERO-EVENT-900, HERO-PLACE-900, HERO-ROCS-900 are three domain modules that refine HERO-EVENT, HERO-PLACE and HERO-ROCS, respectively. They introduce several notions relevant to the history of the 20th century. These modules are an extensible seed for an ontology for the history of the 20th century, and they currently cover the notions needed to describe the events narrated in the archival documents selected for the PRiSMHA project, i.e., the students and workers protest during the years 1968-1969 in Italy. In particular, HERO-EVENT-900 provides the characterization of many specific types of events and states, in several areas, including: *confrontational actions* (e.g., street clashes, police charges, etc.), *protest actions* (e.g., strikes, protest marches, etc.), *life events* (e.g., marriages, dismissals, etc.), *labour-related events* (e.g., labour bargaining, labour agreement achievement, etc.), *phenomena* (e.g., earthquakes, floods, etc.).

HERO-PLACE-900 mainly characterizes notions related (although not exclusively) to modern populated places, such as *city*, *building*, *transportation infrastructure*, etc.

HERO-ROCS-900 offers a set of specific role types (e.g., nationalities, professions, public offices, etc.), a set of specific organization types, in several areas (e.g., in company, healthcare, political, etc. areas) and a set of specific collective entity types (e.g., social classes, nationality and political-based collective entities, etc.). In particular, this module includes a full-fledged characterization of roles (e.g., various types of workers, various types of students) and organization types (e.g., various types of Trade Unions, various types of organizations in the political sphere, and so on) relevant for the domain selected for the PRiSMHA project.

An application version of the HERO ontology (comprising both high-level and domain-specific modules) has been implemented in OWL 2 DL (www.w3.org/OWL). This ontology drives the crowdsourcing system, representing its domain knowledge (see Section 4). The OWL 2 version of the ontology -- particularly its domain-specific modules -- is continually increased; currently, it contains: 424 classes, representing all the entity types that the system knows; 352 properties, which can be used to characterize entities; 79 individuals and nearly 4,500 logical axioms.

4 The Crowdsourcing System: Ontology-driven User Interface

Figure 1 shows the main modules of the PRiSMHA overall system, highlighting the role of the ontology. Digitized documents are "annotated" through the *Crowd-sourcing Platform*, which enables users to build formal semantic descriptions of their content. The user interaction supported by the system is driven by the ontology (*HERO+HERO-900* in the figure), described in Section 3. The ontology also represents the "conceptual vocabulary" of the *Semantic KB*, which contains assertions about domain entities. The Semantic KB is implemented as a RDF triplestore (www.w3.org/RDF), where assertions are expressed by RDF triples. In particular, each assertion is of the form <*s*, *p*, *o*>, where the subject *s* is an entity present in the Semantic KB, *p* is a property (defined in HERO, or belonging to RDF itself – e.g., *rdf:type*) and the object *o* can be either an entity present in the Semantic KB, a literal (i.e., a number, a string, a date, etc.), or a HERO class (e.g., *PhysicalPerson*). The meaning of such a triple is that the 'entity *s* has the value *o* for the property *p*'. For example, the assassination of king Umberto I would be represented as an instance *ev1* of the HERO *Assassination* class, along with a set of assertions including the following ones (whose meaning is straightforward): <*ev1*, *hasAgent*, *GaetanoBresci*>, <*ev1*, *hasPatient*, *UmbertoI*>, <*ev1*, *hasInstrument*, *revolver1*> (where *hasAgent*, *hasPatient*, *and hasInstrument* are HERO properties, and *GaetanoBresci*, *UmbertoI*, and *revolver1* are individuals).[2] Data in the Semantic KB can be made available through a *SPARQL endpoint*, as well as through a navigation User Interface (*Final UI*). Moreover, the Semantic KB (RDF triplestore) can be linked to relevant datasets in the LOD cloud; see, for example, the Zeri and LODE project by Daquino and colleagues (Daquino et al., 2016). These aspects represent a work in progress and are not the focus of the present paper.

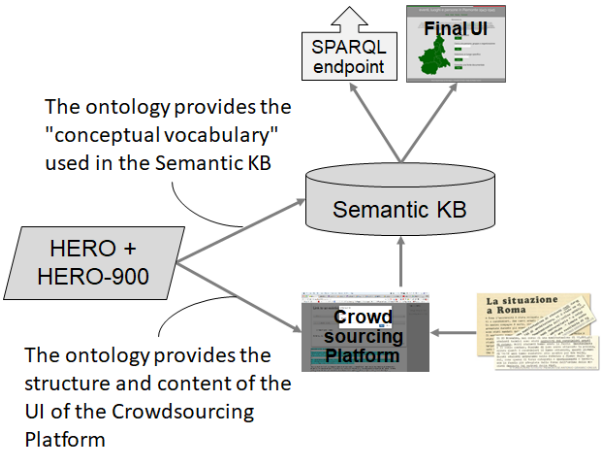

Figure 1: The main modules of the PRiSMHA platform, with the role of the ontology in focus.

The system software architecture is compliant with the classical three-tier model, where the user interface is driven by the application logic which, in turn, manages the interaction with the data layer. In particular, in

[2] Entities, properties, and classes are represented in the triplestore by URI (for example, the entity representing the city of Turin is represented by the following URI: https://w3id.org/prismha/resource/geo/b5e3f1b3-6d4a-4c43-887b-62626516931f). Here we do not use URIs, but simple names, for the sake of readability.

the current prototype of the Crowd-sourcing Platform, the user interface is implemented by exploiting Bootstrap 3.3.7 (getbootstrap.com), Ajax and JQuery 3.3.1 (jquery.com), the application logic is implemented using Spring Boot 1.5.10 (spring.io/projects/spring-boot), and the data are stored into a MySQL 5.6.38 (www.mysql.com) relational database. The interaction with the ontology is performed exploiting OWLAPI 5.1.0 (owlcs.github.io/owlapi) and all the semantic data are stored into an RDF triplestore through Jena TDB 3.6.0 (jena.apache.org/documentation/tdb). The SPARQL endpoint is a work in progress: we are studying the use of Apache Jena Fuseki (jena.apache.org/documentation/fuseki2). The current implementation of the Crowd-sourcing Platform can host different *projects*: each project is associated to a set of documents and to a pool of users (among which there is the *project coordinator*). In the following, we will focus on a single project.

Documents associated to a project can be any kind of archival items; the current prototype manages textual documents, (non-)searchable pdf, and images (jpeg and png). Figure 2 shows some examples.

Figure 2: Examples of archival documents managed by the PRiSMHA platform [*source: (Goy et al., 2019a); copyright: Fondazione Istituto Piemontese Antonio Gramsci*].

Within each document, specific *fragments* can be identified: in textual documents, specific sections of text cab be identified as fragments (in blue in Figure 4); for (non-)searchable pdf there are two options: setting a single fragment corresponding to the whole document, or setting a fragment for each document page; images are currently considered as composed by a single fragment. The goal of the PRiSMHA platform is to provide a full-fledged formal semantic description of the events described in archival documents. The produced formal semantic representations can be seen as a rich stand-off *annotation* of the documents: in particular, each annotation is linked to at least one fragment identified in the document.

In the current version of the platform, there are three types of users, with different privileges: *administrators*, *super-annotators*, and *annotators*. *Administrators* have full privileges, granting them all possible operations, on projects as well as on documents and annotations; *super-annotators* can annotate documents with new semantic representations and can modify existing annotations (also if they have been inserted by other users); simple *annotators* can add new annotations and can propose different representations for the existing ones, but cannot directly modify them. Within each project, each document can be assigned a *supervisor* (typically, a *super-annotator*) and a pool of users, who are enabled to work on that document.

After logging-in, the user can select the project s/he wants to work on. The *project page* (Figure 3) displays the project title, a button to add new documents (*Add new resource*), and the list of documents included in the project: the documents the user is allowed to work on can be clicked on for selection. Some small icons indicate the status of the annotations on each document: an open folder indicates that the document can be annotated (the status can be changed to *closed* by the supervisor: if this is the case, no annotation can be added to the document); a pencil indicates that there are annotations (i.e., semantic representations linked to the document), while a cross indicates that no annotation is present for the current document.

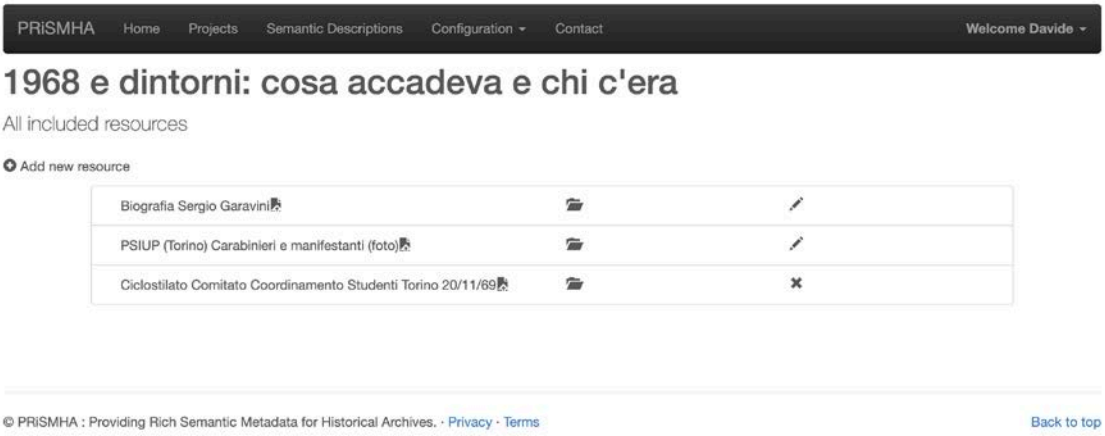

Figure 3: A project page on the PRiSMHA prototype platform.

Figure 4 shows the page corresponding to a textual document (in this case, a biography of Sergio Garavini, a famous Italian Trade Union leader). Users in the pool assigned to the document can identify *fragments* (in blue in the figure), i.e., textual units that will be annotated. By clicking on a specific fragment, users can: (a) See the existing annotations (i.e., semantic representations already linked to that fragment), which are shown on the right-hand bar in the screenshot; by selecting an annotation, users can see the details (event typology, participants, etc.). (b) Add a new annotation, i.e., link the fragment to an entity (typically, an event) and provide a semantic description for it. First, the system suggests to search the system Semantic KB for an already existing entity; if the required entity does not exist, the user can create a new one by clicking on *Add and link new entity* (or on *Add new entity*, if she prefers to link it later on).

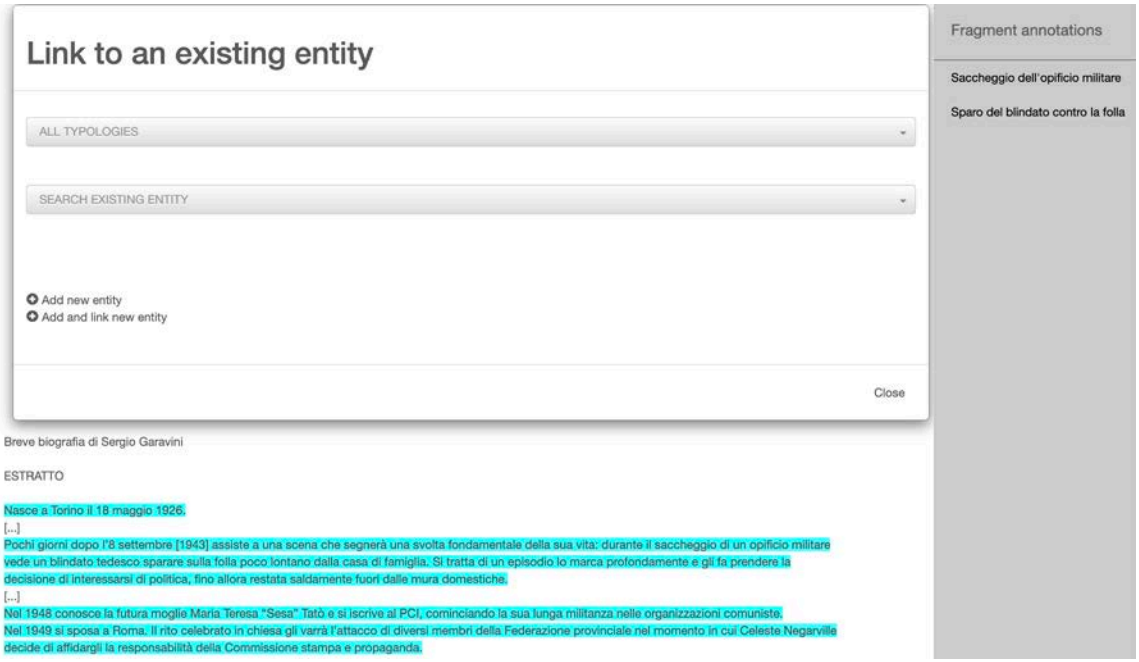

Figure 4: A page on the PRiSMHA platform showing a textual resource (the biography of Sergio Garavini).

Figure 5 shows the current user interface for characterizing the new entity/event by means of its properties. [3] First of all, the user is required to select a typology, i.e., a HERO class (in this case, the user selected *Strike*) and to enter a label for the new entity/event. Depending on the selected ontology class, the system calculates the available properties, according to the ontology (see Section 5.1). For each property, the user is invited to search the Semantic KB (*Search in the TS* [triplestore] *for* …, in the figure), in order to find an entity that can fill that property. For example, if the strike being described (*st1*) took place in Torino, the user can select the entity representing that city (*to*): in this way, the following RDF triple will be added to the triplestore: <*st1*, *hasLocation*, *to*>. If the needed entity is not present in the triplestore, the user can add (and characterize) it.

[3] In the current version of the prototype, labels for the properties are the names defined in the ontology: more user-friendly labels will be defined in the next version of the platform.

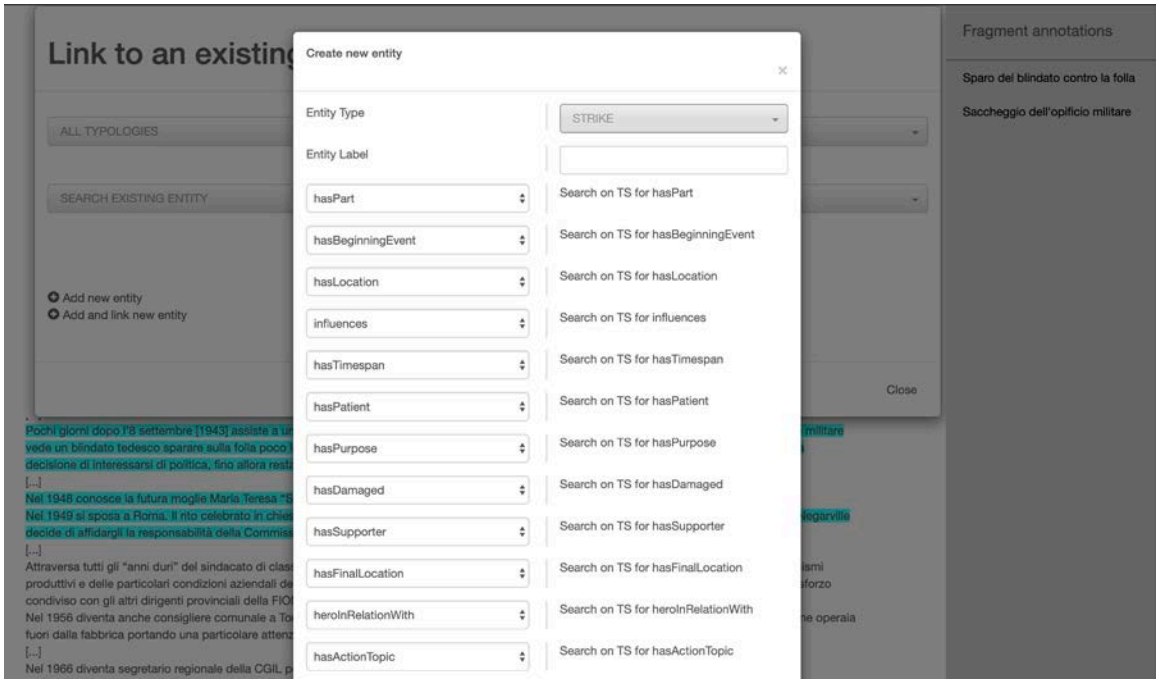

Figure 5: The user interface for characterizing an entity by means of its properties.

As already stated in Section 1, this page highlights one of the most problematic aspects of ontology-driven user interface, i.e., the lack of usability and friendliness of the generated web form: we will discuss our solution in Section 5.

Users can also directly access the Semantic KB, in order to explore the relations among events and entities described in the documents (see Figure 6).

Moreover, as mentioned in Section 1, the PRiSMHA platform enables administrators to set which elements of the ontology should be excluded from the user interface. This is done through a configuration tool that enables administrators to hide classes and properties which "does not make sense" for annotators. For example, in the current configuration, as shown in Figure 7, the platform administrator decided to show concepts such as *GeographicFeature* (i.e., *place*), *PhysicalObject*, *PhysicalPerson*, while she decided to hide concepts -- in red font in the screenshot -- like *SocialObject*, or *CalendarClockIntervalExpression* (used, for example, to characterize expressions denoting particular time intervals). Hidden concepts are not visible in the user interface, but they can be used by the system to build semantic representations, if needed. In the configuration tool, when an item is flagged as *hidden*, all its sub-classes/sub-properties inherit the status (i.e., they are automatically flagged as *hidden*), but they can be selectively modified (i.e., their status can be changed into *shown*).

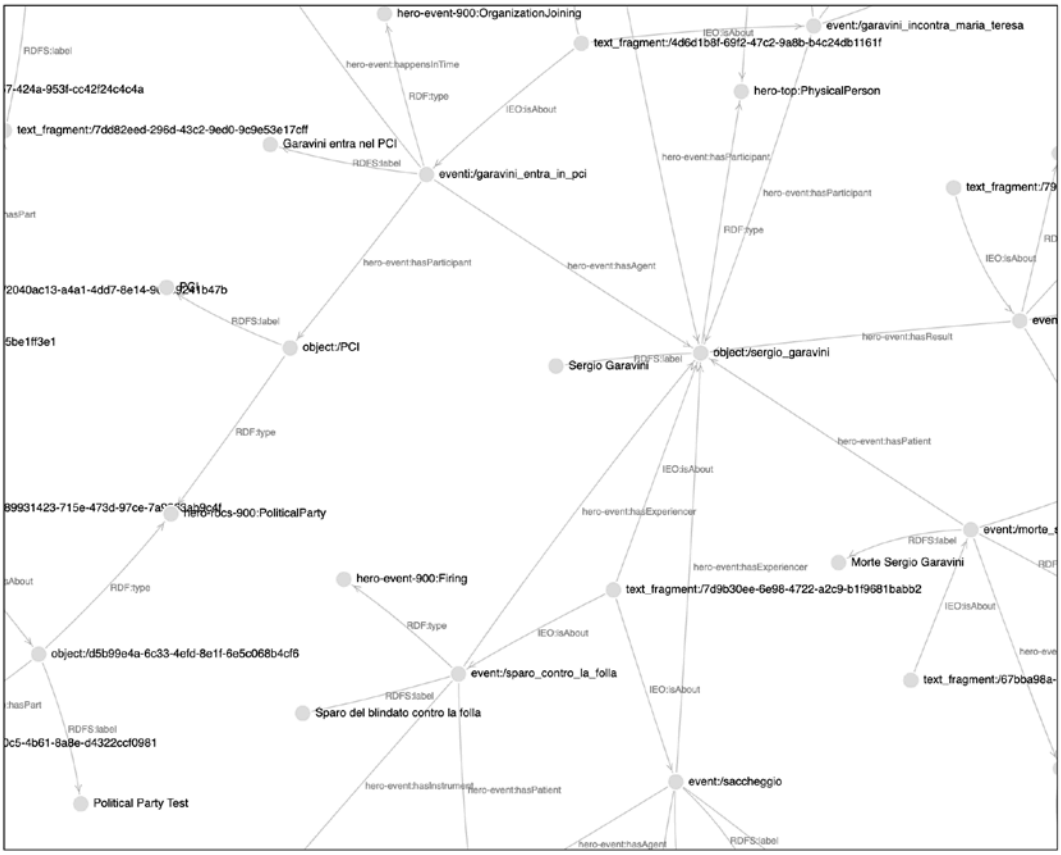

Figure 6: A graphical representation of the knowledge base on the PRiSMHA prototype platform.

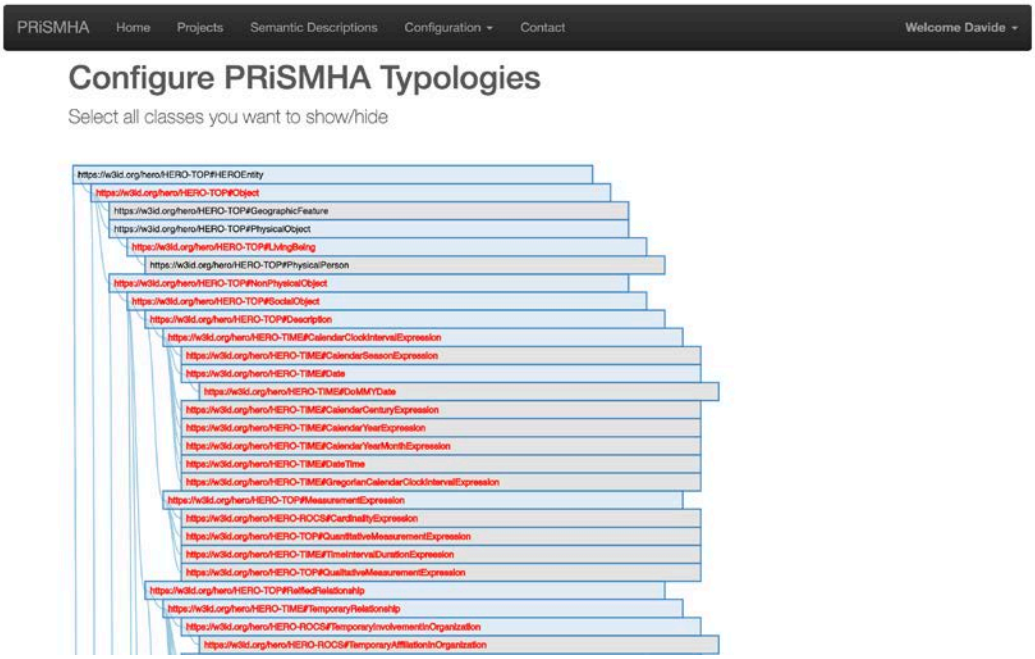

Figure 7: The tool enabling administrators to hide classes, properties, and relations.

In the following section, we will describe the algorithm that we designed and used to extract from the ontology all and only the properties which can be used to characterize an event of a given type. Moreover, we will present the results of a user study aimed at providing us with suitable strategies to rank, group, and present the event properties in the user interface.

5 The Problem: Selecting Properties from the Ontology to Characterize Events

5.1 The Algorithm Extracting from the Ontology the Properties Available for Characterizing Events

As stated above (see Section 4), the HERO ontology plays a central role in guiding the user interaction. Users may add new entity descriptions (e.g., descriptions of events, persons, organizations, etc.) to the system Semantic KB or they may modify descriptions already present in it. Each entity must be described in terms of the ontology, which means that, for each entity, its type -- i.e., the HERO class it belongs to -- and a set of features -- i.e., assertions involving HERO properties -- should be specified. Within the Semantic KB, assertions are expressed by RDF triples. In general, any property offered by the ontology is suitable for describing entities of some (but not necessarily all) classes. For instance, all properties representing thematic roles (*hasAgent*, *hasPatient*, etc.) are suitable for describing only events and states (in the specific sense that only events and states can be subjects of triples mentioning thematic roles), but not other kinds of entities: It would make no sense to state, say, that an organization has some entity playing a thematic role in it (in other words, an instance of the HERO *Organization* class can never be the subject of a triple mentioning a thematic role). Even restricting our attention to events and thematic roles, we find some incompatibilities. For instance, in a historical domain, it would make no sense specifying an agent who intentionally carried out an earthquake event (i.e., an instance of the HERO *Earthquake* class can never be subject in a triple mentioning the HERO *hasAgent* property, even though an earthquake is a specific kind of event and the *hasAgent* property specifies a thematic role).

It is worth noting that we assume that the ontology correctly specifies the mentioned incompatibilities, in the sense that it logically implies them. However, we cannot assume that the ontology explicitly represents each one of them by a single axiom of a specific kind. Thus, discovering these incompatibilities is an unavoidable need, typically requiring a (possible complex) reasoning on the ontology.

Moreover, besides the compatibility between HERO classes and HERO properties, we should also consider the issue of the relevance of properties w.r.t. classes. In fact, given a class in the ontology, the properties compatible with the class could be not equally relevant for describing the instances of the class. For example, we can expect the HERO *hasPatient* property (expressing the entities affected by an event) being usually more relevant than the HERO *hasForce* property (expressing the non-volitional causer of the event) for describing an earthquake in the historical domain. This latter issue is the subject of the user study discussed in Section 5.2, while in this section we discuss the former one, i.e. the problem of singling out the compatible properties for each class in the ontology.

In this context, we say that an ontological property *p* is compatible with an ontological class *C*, if and only if there exists at least one assertion of the form $<c_1, p, o>$ (where c_1 is an instance of *C*), which is consistent with the ontology. In the present work, we make the common assumption that the ontology is expressed as a Description Logic theory (Baader et al., 2010).[4] In such a framework, we can formalize our notion of compatibility by reducing it to that of concept satisfiability in Description Logics (Baader et al., 2010), as follows:

Definition 1: Given an ontology *O*, a named class *C* and a property *p*, we say that *p* is compatible with *C* in *O* (i.e. the pair (*C*, *p*) is valid in O) iff the concept $C \sqcap (\exists p.\top)$ is satisfiable with respect to *O*.

[4] Since the application version of the HERO ontology is expressed as an OWL 2 DL theory, the results discussed in this section also apply to it.

Given the notion of concept satisfiability in Description Logics, the definition above states that the pair (*C*, *p*) is valid in the ontology O if and only if there exists a model of O (i.e., an interpretation satisfying all the axioms in O) in which the concept $C \sqcap (\exists p.\top)$ is nonempty. Since the mentioned concept represents the set of instances of the *C* class that have some value for the property *p*, it should be clear that the definition above properly formalizes our notion of compatibility between classes and properties in an ontology.
Singling out the valid pairs (*C*,*p*) in an ontology is a complex task, involving formal reasoning on the ontology, that we cannot delegate to users. Instead, the system should provide the users with the valid pairs. It is well known that the standard reasoning problems (including concept satisfiability) are largely intractable (N2ExpTime) in SROIQ (Kazakov, 2008), the Description Logic underlying OWL 2. Moreover, in an ontology with *m* named classes and *n* named properties, the number *m*n* of the possible class-property combinations could be very large (in the current version of the HERO ontology, there are nearly 150,000 of such combinations). Therefore, it could be unfeasible to detect the valid pairs by explicitly testing the satisfiability of the concepts $C \sqcap (\exists p.\top)$, for each named class *C* and each property *p*, even using a best-performing SROIQ reasoner (Parsia et al., 2017).
In the following, we present an algorithm that computes the set of valid pairs (*C*, *p*) and exploits the axioms present in the ontology to reduce the number of reasoner invocations for testing the satisfiability of the corresponding concepts $C \sqcap (\exists p.\top)$, for each named class *C* and each property *p* in the ontology.
The algorithm is based on the following properties:

Property 1: Given two named classes C_1 and C_2, and a property *p* in an ontology *O*, if C_1 and C_2 are disjoint and the domain of *p* is either C_1 or a subclass of C_1, then (C_2, p) is not a valid pair in *O*.

Proof: Let us suppose that (C_2, p) is a valid pair in *O*. Then, by Definition 1, the concept $C_2 \sqcap (\exists p.\top)$ is satisfiable with respect to *O*; this means that there exists a model of *O* in which an instance *x* of C_2 has some value for the property *p*; given that, by hypothesis, the domain of *p* is either C_1 or a subclass of C_1, it follows that *x* is an instance of C_1 in that same model of *O*, but this contradicts the hypothesis that C_1 and C_2 are disjoint.

Property 2: Given a named class C, and a satisfiable property *p* in an ontology *O*, if C is either the domain of *p* or a superclass of it, then (C, p) is a valid pair in *O*.

Proof: By hypothesis, *p* is a satisfiable property, which means that there exists a model of *O* in which an entity *x* has some value for the property *p*; but, by hypothesis, C is either the domain of *p* or a superclass of it, therefore *x* is an instance of C in that same model of *O*; this means that in that model *x* is an instance of the concept $C \sqcap (\exists p.\top)$, which means that such concept is satisfiable w.r.t. O; therefore, by Definition 1, (C, p) is a valid pair in *O*.

Property 3: Given two named classes C_1 and C_2, and a property *p* in an ontology *O*, if (C_1, p) is a valid pair in *O* and C_2 is a superclass of C_1, then (C_2, p) is a valid pair in *O*.

Proof: By hypothesis, (C_1, p) is a valid pair in *O*, thus, by Definition 1, the concept $C_1 \sqcap (\exists p.\top)$ is satisfiable with respect to *O*, which means that there exists a model of *O* in which an entity *x* is an instance of the mentioned concept, i.e. *x* is an instance of C_1 and it has some value for the property *p*; given that, by hypothesis,C_2 is a superclass of C_1, we have that *x* is an instance of C_2 in that same model of *O*; therefore, the concept $C_2 \sqcap (\exists p.\top)$ is satisfiable with respect to *O*, thus, by Definition 1, (C_2, p) is a valid pair in *O*.

Property 4: Given a named class C and two properties p_1 and p_2 in an ontology *O*, if (C, p_1) is a valid pair in *O* and p_2 is a superproperty of p_1, then (C, p_2) is a valid pair in *O*.

Proof: By hypothesis, (C, p_1) is a valid pair in *O*, thus, by Definition 1, the concept $C \sqcap (\exists p_1.\top)$ is satisfiable with respect to *O*, which means that there exists a model of *O* in which an entity *x* is an instance of the mentioned concept, i.e. *x* is an instance of C and it has some value for the property p_1; given that, by hypothesis, p_2 is a superproperty of p_1, we have that *x* has some value for the property p_2 in that same

model of *O*; therefore, the concept $C \sqcap (\exists p_2.\top)$ is satisfiable with respect to *O*, thus, by Definition 1, (C, p_2) is a valid pair in *O*.

Property 5: Given two named classes C_1 and C_2, and a property p in an ontology *O*, if (C_1, p) is not a valid pair in *O* and C_2 is a subclass of C_1, then (C_2, p) is a not a valid pair in *O*.

Proof: Let us suppose that (C_2, p) is a valid pair in *O*. Then, by Definition 1, the concept $C_2 \sqcap (\exists p.\top)$ is satisfiable with respect to *O*; this means that there exists a model of *O* in which an instance *x* of C_2 has some value for the property *p*; given that, by hypothesis, C_2 is a subclass of C_1, it follows that that *x* is an instance of C_1 in that same model of *O*; therefore, the concept $C_1 \sqcap (\exists p.\top)$ is satisfiable w.r.t. O; thus, by Definition 1, (C_1, p) is a valid pair in *O*, but this contradicts the hypothesis.

Property 6: Given a named class C and two properties p_1 and p_2 in an ontology *O*, if (C, p_1) is not a valid pair in *O* and p_2 is a subproperty of p_1, then (C, p_2) is not a valid pair in *O*.

Proof: Let us suppose that (C, p_2) is a valid pair in *O*. Then, by Definition 1, the concept $C \sqcap (\exists p_2.\top)$ is satisfiable with respect to *O*; this means that there exists a model of *O* in which an instance *x* of C has some value for the property p_2; given that, by hypothesis, p_2 is a subproperty of p_1, it follows that *x* has some value for the property p_1 in that same model of O; therefore, the concept $C \sqcap (\exists p_1.\top)$ is satisfiable w.r.t. O; thus, by Definition 1, (C, p_1) is a valid pair in *O*, but this contradicts the hypothesis.

The algorithm can be sketched as follows:

ALGORITHM 1: Computation of Valid Pairs (class, property)

```
1  Set computeValidPairs(Ontology ont){
2      Set validPairs = ∅;
3      Set classes = {C: C is a named class in ont};
4      Set props = {p: p is a named property in ont};
5      Set pairsToCheck = {(C,p): C ∈ classes ∧ p ∈ props};
6      foreach(p in props)
7          p.doms = retrieveNamedDomains(ont,p);
8      foreach(C in classes){
9          Set disjC = retrieveNamedDisjointClasses(ont,C) ;
10         foreach(p in props)
11             //On the basis of Property 1…
12             if(disjC ∩ p.doms ≠ ∅)
13                 pairsToCheck = pairsToCheck − {(C,p)};
14     }
15     //On the basis of Property 2 and assuming that each property in ont is satisfiable…
16     foreach(p in props)
17       foreach(C in p.doms){
18         pairsToCheck = pairsToCheck − {(C,p)};
19         validPairs = validPairs ∪ {(C,p)};
20       }
21     while(pairsToCheck ≠ ∅){
22         select (C,p) in pairsToCheck;
23         if((C ⊓ ∃p.⊤) is satisfiable w.r.t. ont){
24             //On the basis of Properties 3 and 4…
25             Set newValidPairs_(C,p) = {(D,q): (D,q) ∈ pairsToCheck ∧
                                          (D ≡ C ∨
                                          D is a named class, which is asserted as a direct or
                                          indirect superclass of C in ont)
                                          ∧ (q ≡ p ∨
                                          q is a named property, which is asserted as a
                                          direct or indirect superproperty of p in ont)};
26             pairsToCheck = pairsToCheck − newValidPairs;
27             validPairs = validPairs ∪ newValidPairs;
```

28 } else {
29 //On the basis of Properties 5 and 6…
30 Set $newInvalidPairs_{(C,p)} = \{(D,q): (D,q) \in pairsToCheck \wedge$
$(D \equiv C \vee$
$D\ is\ a\ named\ class, which\ is\ asserted\ as\ a\ direct\ or$
$indirect\ subclass\ of\ C\ in\ ont) \wedge$
$(q \equiv p \vee$
$q\ is\ a\ named\ property, which\ is\ asserted\ as\ a$
$direct\ or\ indirect\ subproperty\ of\ p\ in\ ont)\}$;
31 $pairsToCheck = pairsToCheck - newInvalidPairs$;
32 }
33 }
34 return $validPairs$;
35 }

After some variable initializations (rows 1-5), in which, in particular, the set of all possible class-property combinations is computed (row 5), the specified mechanism works in two phases. The first phase (rows 6-20), reduces the set of class-property pairs to check essentially by browsing the ontology and without invoking any complete Description Logic reasoner. The second phase (rows 21-33), checks the validity of some pairs of the form (C, p), by verifying the satisfiability of the corresponding concepts $C \sqcap \exists p.\top$, on the basis of Definition 1 (row 23). Then, it exploits each result of such a check to derive either the validity or the invalidity of other pairs, by browsing the ontology and without invoking any complete Description Logic Reasoner (rows 24-32).

Let us analyze the two phases in detail. The first phase consists of two steps. In the first step (rows 6-14), a set of pairs are recognized as invalid. In particular, for each property p, the algorithm browses the ontology (through the invocation of the function $retrieveNamedDomains(\dots)$, reported and explained below) for retrieving all the named classes that are either specified as domains of p or as superclasses of domains of p (rows 6-7). Then, for each class C, it computes the set of named classes that can be recognized as disjoint from C by only browsing the ontology (rows 8-9, where the invocation of the function $retrieveNamedDisjointClasses(\dots)$ -- reported and explained below -- performs the actual browsing of the ontology). If there is at least one class that is both disjoint with C and either a domain of a property p or a superclass of a domain of p (row 12), then, by Property 1, the pair (C, p) is invalid, thus the algorithm removes it from the pairs to check (row 13). In the second step of the first phase (rows 15-20), a set of pairs are recognized as valid. The algorithm assumes that each property p in the ontology is satisfiable (i.e. that there exists a model of the ontology in which two entities are related by the property p). Therefore, by Property 2, the algorithm inserts each pair (C, p), where C is either a domain of p or a superclass of it, into the set of valid pairs, without the need of explicitly check its validity (rows 18 and 19).

In the second phase, the algorithm selects a pair (C, p) among those that should be checked for validity (row 22). Then, it invokes a complete Description Logic reasoner in order to verify whether the corresponding concept $(C \sqcap \exists p.\top)$ is satisfiable (row 23). If it is, by Definition 1, the selected pair is valid. Therefore, by Properties 3 and 4, the algorithm, besides the validity of (C, p) and without any Description Logic complete reasoner invocation, it asserts also the validity of any pair (D, q), such that in the ontology D is asserted as a direct or indirect superclass of C or q is asserted as a direct or indirect superproperty of p (rows 25-27). On the opposite, if the checked concept is unsatisfiable, by Definition 1, the selected pair is invalid. Therefore, by Properties 5 and 6, the algorithm, besides the invalidity of (C, p) and without any Description Logic complete reasoner invocation, it asserts also the invalidity of any pair (D, q), such that in the ontology D is asserted as a direct or indirect subclass of C or q is asserted as a direct or indirect subproperty of p (rows 30-31).

The choice of the pair (C, p) in row 22 deserves a few words. It should be clear that the presented mechanism aims at reducing the invocations of any Description Logic complete reasoner in order to check the validity of a set of class-property pairs. To this aim, in the second phase of the algorithm, the result of the satisfiability check relevant to the selected pair is exploited to derive the (un)satisfiability of other pairs. Therefore, a good

criterion for selecting the pair (C, p) in row 22 could be the one which selects the pair that maximizes the number of solved pairs with one single satisfiability check, i.e.:

$$(C, p) = \underset{\{x \in pairsToCheck\}}{\operatorname{argmax}} \left(P(x\ is\ valid) * |newValidPairs_x| + \left(1 - P(x\ is\ valid)\right) * |newInvalidPairs_x|\right),$$

where the sets $pairsToCheck$, $newValidPairs_x$ and $newInvalidPairs_x$ are specified in the algorithm above. The problem here is estimating the probability that a pair is valid. In the current implementation of the algorithm, we uniformly estimated $P(x\ is\ valid) = \frac{6}{7}$.

The following algorithm retrieves the named class in the ontology that are specified as domains of a property p or as superclasses of such classes, by only browsing the ontology and without invoking any complete Description Logic reasoner:

ALGORITHM 2: Retrieval of Named Domains

1 Set **retrieveNamedDomains(Ontology ont, NamedProperty p)**{
2 Set $domsP = \{C: C\ is\ a\ named\ class\ in\ ont \wedge C\ is\ asserted\ as\ a\ domain\ of\ p\ in\ ont\}$;
3 Set $domsP_2 = \{C_2: C_2\ is\ a\ named\ class\ in\ ont\ \wedge$
$(\exists C)(C \in domsP \wedge C_2\ is\ asserted\ as\ a\ direct\ or\ indirect\ superclass\ of\ C\ in\ ont)\}$;
4 Set $superPropsP = \{q: q\ is\ a\ named\ property\ in\ ont \wedge$
$q\ is\ asserted\ as\ a\ direct\ or\ indirect\ superproperty\ of\ p\ in\ ont\}$;
5 Set $domsQ = \{C_3: C_3 is\ a\ named\ class\ in\ ont \wedge$
$(\exists q)(q \in superPropsP \wedge C_3 is\ asserted\ as\ a\ domain\ of\ q\ in\ ont)\}$;
6 Set $domsQ_2 = \{C_4: C_4\ is\ a\ named\ class\ in\ ont \wedge$
$(\exists C_3)(C_3 \in domsQ \wedge C_4\ is\ asserted\ as\ a\ direct\ or\ indirect\ superclass\ of\ C_3\ in\ ont)\}$;
7 Set $invPropsP = \{r: r\ ia\ a\ named\ property\ in\ ont \wedge$
$r\ is\ asserted\ as\ an\ inverse\ property\ of\ p\ in\ ont\}$;
8 Set $rangesR = \{C_5: C_5\ is\ a\ named\ class\ in\ ont \wedge$
$(\exists r)(r \in invPropsP \wedge C_5\ is\ asserted\ as\ a\ range\ of\ r\ in\ ont)\}$;
9 Set $rangesR_2 = \{C_6: C_6\ is\ a\ named\ class\ in\ ont \wedge$
$(\exists C_5)(C_5 \in rangesR \wedge C_6\ is\ asserted\ as\ a\ direct\ or\ indirect\ superclass\ of\ C_5 in\ ont)\}$;
10 Set $superPropsInvPropsP = \{s: s\ ia\ a\ named\ property\ in\ ont \wedge$
$(\exists\ r)(r \in\ invPropsP \wedge$
$s\ is\ asserted\ as\ a\ direct\ or\ indirect\ superproperty\ of\ r\ in\ ont)\}$;
11 Set $rangesS = \{C_7: C_7\ is\ a\ named\ class\ in\ ont \wedge$
$(\exists s)(s \in superPropsInvPropsP \wedge C_7\ is\ asserted\ as\ a\ range\ of\ s\ in\ ont)\}$;
12 Set $rangesS_2 = \{C_8: C_8\ is\ a\ named\ class\ in\ ont \wedge$
$(\exists C_7)(C_7 \in rangesS \wedge C_8\ is\ asserted\ as\ a\ direct\ or\ indirect\ superclass\ of\ C_7\ in\ ont)\}$;
13 $domsP = domsP \cup domsP_2 \cup domsQ \cup domsQ_2 \cup rangesR \cup rangesR_2 \cup rangesS \cup rangesS_2$;
14 return $domsP$;
15 }

The returned set of named classes contains all those classes that are explicitly specified as domains of the property p (row 2), as well as the direct and indirect superclasses of such classes (row 3). Moreover, the domains of a superproperty of p (rows 4-5), as well as the superclasses of such domains (row 6), are (superclasses of) domains of p, thus they must be included, too. Since the ranges of the inverse properties of p (rows 7-8), as well as the superclasses of some ranges (row 9), are (superclasses of) domains of p, the algorithm collects them, too. Similarly, the ranges of the superproperties of the inverse properties of p (rows 10-11), as well as the superclasses of such ranges (row 12) must be collected, too. It is worth noting that the algorithm performs a sort of incomplete reasoning, browsing the class and the property hierarchies and searching for explicitly asserted domains and ranges.

The following algorithm retrieves the named classes that are specified as disjoint with a class C:

ALGORITHM 3: Retrieval of Named Disjoint Classes

1 Set **retrieveNamedDisjointClasses(Ontology ont, NamedClass C)**{
2 Set $disjC = \{D: D\ is\ a\ named\ class\ in\ ont \wedge D\ is\ asserted\ as\ disjoint\ with\ C\ in\ ont\}$;
3 Set $disjC_2 = \{D_2: D_2\ is\ a\ named\ class\ in\ ont \wedge (\exists D)(D \in disjC \wedge D_2\ is\ asserted\ as\ a\ direct\ or\ undirect\ subclass\ of\ D\ in\ ont)\}$;
4 Set $disjC_3 = \{D_3: D_3\ is\ a\ named\ class\ in\ ont \wedge (\exists C_2)(C_2\ is\ a\ named\ class\ in\ ont \wedge C_2 is\ asserted\ as\ a\ direct\ or\ undirect\ superclass\ of\ C\ in\ ont \wedge C_2\ is\ asserted\ as\ disjoint\ with\ D_3\ in\ ont)\}$;
5 Set $disjC_4 = \{D_4: D_4\ is\ a\ named\ class\ in\ ont \wedge (\exists D_3)(D_3 \in disjC_3 \wedge D_4\ is\ asserted\ as\ a\ direct\ or\ undirect\ subclass\ of\ D_3\ in\ ont)\}$;
6 $disjC = disjC \cup disjC_2 \cup disjC_3 \cup disjC_4$;
7 return $disjC$;
8 }

The returned set of named classes contains all those classes that are explicitly asserted as disjoint with C (row 2). If a class is disjoint with C, so are its subclasses (row 3). Moreover, if a class is asserted as disjoint with a superclass of C, it is also disjoint form C (row 4) and so are its subclasses (row 6). Also in this case, the algorithm performs a sort of incomplete reasoning, browsing the class hierarchy and searching for explicitly asserted disjoint classes.

Algorithm 1 eventually returns the set of valid class-property pairs, given the input ontology. For each class *C* in the ontology, the set $\{(C, p_1), \ldots (C, p_n)\}$ of valid pairs actually expresses a constraint on the RDF triples in the Semantic KB. Such a constraint states that any instance *x* of *C* can only be subject of triples of the form <*x*, *p*, *o*>, where $p \in \{p_1, \ldots p_n\}$ (besides triples of the form <*x*, *q*, *o*>, where *q* is a pre-defined property in RDF, RDFS, or OWL such as *rdf:type*, *rdfs:label*, etc.). The mentioned constraint can be represented by the SHACL (SHACL, 2017)[5] following *shape*:[6]

c-shape
a sh:NodeShape ;
sh:targetClass C ;
sh:closed true ;
sh:ignoredProperties (rdf:type, rdfs:label, …) ;
sh:property [
sh:path p_1 *;*
] ;
…
sh:property [
sh:path p_n *;*
] .

As stated in Section 3, the current version of the HERO ontology contains 424 classes and 352 properties, which generate a space of 149,248 class-property combinations, in which valid pairs have to be singled out. The algorithm above allowed us to compute the set of valid pairs by only 12,434 invocations of a complete Description Logic reasoner. Thus, the presented algorithm saved nearly 92% of the reasoner invocations, w.r.t. a brute-force approach performing a reasoner invocation for each class-property combination. By exploiting the Konclude complete OWL 2 reasoner (Steigmiller et al., 2014) -- which turned out to be the best-performing reasoner in the Owl Reasoner Evaluation 2015 Competition (Parsia et al., 2017) -- the whole computation took 3,811 seconds (i.e., a little more than one hour) on a Windows 7 Enterprise, SP 1, 16 GB

[5] SHACL is a W3C Recommendation defining a language for validating RDF graphs against a set of constraints, typically provided as constructs called *shapes*.

[6] *sh* is the prefix for the namespace http://www.w3.org/ns/shacl#.

RAM PC. We also ran a brute-force algorithm on the same PC, with a four hours timeout. Such an algorithm timed-out after 39,600 reasoner invocations, i.e., it checked only 26,5% of the class-property combinations before reaching the timeout.

5.2 User Study

The user interface based on the output of the algorithm described in Section 5.1 currently provides users with a flat sequence of properties (see Figure 5), and it could be significantly improved by taking into account the relevance of the different properties that can be used to characterize an event of a given type. To this end, we designed a user study aimed at eliciting users opinion about the relevance of each property that can be used to characterize a specific type of event.

Running the algorithm described in Section 5.1, we noticed that the large majority of classes in HERO-900 referring to domain-dependent event typologies can be characterized by (almost) the same set of properties. Moreover, by analyzing the results of the ontological analysis (Goy et al., 2019a) that led to the class taxonomy of HERO-900, we identified the following major semantically homogeneous groups of classes:

- **Group 1**: Classes referring to physical confrontations (32 classes).
- **Group 2**: Classes referring to protest actions (30 classes).
- **Group 3**: Classes referring to life events (27 classes).
- **Group 4**: Classes referring to labour-related events (18 classes).
- **Group 5**: Classes referring to communicative acts (17 classes).
- **Group 6**: Classes referring to (natural) phenomena (1 class).

We decided to focus on groups 1-3, as they emerged to be the most meaningful with respect to the domain, on the basis of both the number of classes they include, and the results of the domain analysis performed by historians on the archival documents; see (Goy et al., 2019a). Such an analysis also enabled us to select the 7 most representative classes for each group, on the basis of the occurrence of events of these types in the documents:[7]

- **Group 1**: *PhysicalConfrontation, Beating, StreetClash, PoliceCharge, Firing, KillingOfPeople, Wounding*.
- **Group 2**: *ProtestAction, Demonstration, Strike, Picketing, ProtestMarch, PlaceOccupation, SitIn*.
- **Group 3**: *PersonBirth, PersonDeath, Marriage, RoleAssumption, Hiring, Dismissal, Retiring*.

Instances of all classes in these groups can be characterized by means of the following list of properties: *hasActionTopic, hasAdvantaged, hasAgent, hasBeginningEvent/isBeginningEventOf, hasCausalFactor/isCausalFactorOf, hasDamaged, hasEndingEvent/isEndingEventOf, hasExperiencer, hasFinalLocation, hasFinalTime, hasForce, hasInitialLocation, hasInitialTime, hasInstrument, hasIntermediateLocation, hasLocation, hasOpponent, hasParticipant, hasPatient, hasProspectiveParticipant, hasPurpose, hasResult, hasRetaliation/isRetaliationFor, hasSubEvent/isSubEventOf, hasSupporter, hasTheme, hasTimespan, heroInRelationWith, Influences/isInfluencedBy*.

Moreover, in Group 3 there are two classes which admit two additional, more specific, properties, namely: *hasSpouse* (only for *Marriage*), *hasAssumedRole* (only for *RoleAssumption*).

We recruited 30 participants, selected among potential users of the PRiSMHA platform, and assigned 10 of them to each group. For each group: (i) 50% of users were males and 50% were females; (ii) The average age was 39; (iii) 40% of users were expert in History or Human Studies, while 60% of participants had a different background (computer scientists, administrative staff, science students), but all of them were interested in historical and cultural topics.

[7] The selection of 7 classes for each group under study aimed at reducing the work overload imposed to the participants of the user study.

We designed three online questionnaires in which, after some profile information, users had to rate, on a 1 (definitely irrelevant) to 5 (extremely important) scale, the relevance of each property for describing events belonging to each event typology (class). For example, in Group 1, for the property *hasAgent*, the users had to answer the following question (the original questionnaire was in Italian: We here provide the English translation for the sake of the reader):

*The **agent** is the participant who voluntarily performs the action. Please, notice that an agent can be a person, a group of persons, but also an organization (e.g., a State, a political party, a university, a company, etc.). For instance, "Fiat Chrysler dismissed 10 workers": Fiat Chrysler is the agent.*
On a 1-to-5 scale (1=definitely irrelevant, 5=extremely important) evaluate the importance of being able to specify the agents in events of the following types:

Physical confrontation	1 2 3 4 5
Beating	1 2 3 4 5
Street clash	1 2 3 4 5
Police charge	1 2 3 4 5
Firing	1 2 3 4 5
Killing of people	1 2 3 4 5
Wounding	1 2 3 4 5

Moreover, for each property, users could add a free comment. The questionnaires ended with a space for a global free comment.
Before starting, users were invited to have a look at some examples of archival documents (see, for instance, Figure 2) and to read a short, non-technical explanation of the intended meaning of each involved class (e.g., ***Firing**: It represents those confrontation actions where people, collectives, or organizations -- e.g., Police, States, etc. -- fire at people or objects, or where people, collectives, or organizations fire at each other. It is possible to fire using different instruments: shotguns, tanks, rockets, bombs, etc.*).

5.3 Results and Discussion

Table 1a and Table 2a (Appendix A.1) report, respectively, the mean and the standard deviation values for Group 1. Interesting hints come from free comments provided by users. In particular, in some cases, the disagreement (but also a low relevance score) seems to arise from a difficulty in catching the intended meaning of the property (in particular, with respect to the given event typologies), or from giving it different interpretations; for example:

- *hasExperiencer*: some users wrote that the meaning of this property is unclear, although the relevance score for it is quite high (mean between 3 and 4).
- *hasTheme*: many users wrote that the meaning of this property is unclear; moreover, this property is the only one that obtained a very low relevance score (lower than 3).
- *hasProspectiveParticipant*: it seems that users provide different interpretations for this property (e.g., people who have the intention to participate, people who are involved against their will, people called for a meeting, etc.); besides disagreement between participants, this property also obtained heterogeneous relevance scores for the different event typologies.
- *hasForce*: some users wrote that this property can be confused with *hasCausalFactor/isCausalFactorOf*; also this property, besides disagreement between participants, obtained heterogeneous relevance scores for the different event typologies.

These comments suggest us to review the actual definition of these properties in HERO, a task we scheduled for next version of our prototype (see Section 6).
An interesting correlation we can observe is the following: for properties that obtained a high relevance score, users tend to agree, while for properties that obtained a lower relevance score, users tend to disagree. This could mean that the relevance of some properties is out of discussion, while the importance of other properties

varies for different users: for somebody some properties are relevant, while for others they are not so important. This implies that no property can be excluded tout-court (and this is reasonable), and the user interface should enable users to access any property, if they think it is relevant (see Section 5.4).
We can also notice that properties that can be considered as semantically "related" (analogous, symmetrical) do not always obtain the same scores; this is the case for the following properties:

- *hasTimespan* and *hasInitialTime* (mean relevance greater than 4 for all classes; moderate agreement), *hasFinalTime* (mean relevance greater than 4 for all classes; different scores for the st. dev., depending on the class).
- *hasLocation* (mean relevance greater than 4 for all classes, fundamental agreement), *hasFinalLocation* and *hasInitialLocation* (mean relevance greater than 4 for some classes, but between 3 and 4 for other classes; moderate disagreement), *hasIntermediateLocation* (mean relevance between 3 and 4 for some classes, but lower than 3 for other classes; substantial disagreement).
- *hasDamaged* (mean relevance greater than 4 for all classes; moderate agreement) and *hasAdvantaged* (mean relevance between 3 and 4 for all classes; moderate agreement).
- *hasOpponent* (mean relevance greater than 4 for some classes, but between 3 and 4 for other classes; moderate agreement) and *hasSupporter* (mean relevance between 3 and 4 for all classes; moderate agreement).
- *hasBeginningEvent/isBeginningEventOf* (mean relevance greater than 4 for some classes, but between 3 and 4 for other classes; substantial disagreement) and *hasEndingEvent/isEndingEventOf* (mean relevance between 3 and 4 for all classes; substantial disagreement).

Table 3a and Table 4a (Appendix A.1) report, respectively, the mean and the standard deviation values for Group 2. The same correlations noticed for Group 1 can be seen also for Group 2 (for properties that obtained a high relevance score, users tend to agree, while for properties that obtained a lower relevance score, users tend to disagree): this fact strengthen the recommendation for the user interface mentioned above, i.e., the user interface should enable users to access any property, if they think it is relevant.
The major differences with respect to Group 1 are the following:

- *hasActionTopic* obtains a **much better** result.
- *hasExperiencer* and *hasTheme* obtain a **better** result.
- The following properties obtain a **slightly better** result: *hasDamaged, hasPurpose, hasSupporter, hasProspectiveParticipant*.
- The following properties obtain a **slightly worse** result: *hasForce, hasRetaliation/isRetaliationFor*.
- *hasInstrument* and *hasSubEvent/isSubEventOf* obtain the **same result** in terms of relevance, but with **more agreement**; *Influences/isInfluencedBy* obtains the **same result** in terms of relevance, but with **less agreement**.

User comments can justify some better results:

- *hasProspectiveParticipant*: some users wrote that this property is important for evaluating the success of protest actions (like demonstrations, strikes, protest marches, etc.).
- *hasDamaged*: some users wrote that, in this kind of actions, they found this property sometimes similar to the *hasPatient*.

Moreover, given the semantic context of protest actions, the slightly better result obtained by *hasPurpose* and *hasSupporter* (typical properties of protest actions) is not surprising. The same semantic context clearly explains why *hasActionTopic* obtained a much better result; protest actions, in fact, are typically characterized by their "topic", given that, in HERO this property has the following intended meaning (explained to the user study participants):

hasActionTopic*: It associates an action to its "topic"; in fact, some types of action are typically "about a topic", like, for instance: conferences, speeches, meetings, debates, etc. (a conference about climate changes, a debate about Che Guevara, and so on).*

The slightly better result in Group 1 obtained by *hasForce*, *hasRetaliation/isRetaliationFor* could be explained by the fact that these properties seems to be quite important in the characterization of physical confrontational actions. The better results obtained in Group 2 by *hasExperiencer* and *hasTheme* are more difficult to explain, but the concepts of *experiencer* and *theme* were probably more difficult to grasp, and this could explain at least the fact that users have different opinions about them

Table 5a and Table 6a (Appendix A.1) report, respectively, the mean and the standard deviation values for Group 3. These results clearly show that Group 3 deserves a different analysis, since they are quite different with respect to Group 1 and 2. The major difference is the very high disagreement among users: there are a lot of properties with a quite low relevance score and a substantial disagreement, while no property (but the two class-specific ones) obtained a high relevance score with a fundamental agreement. Moreover, a lot of properties obtained very different relevance scores (ranging from 2 up to 4,6) for the different classes involved. Finally, many properties obtained a different relevance score with respect to those obtained in Group 1 and Group 2. User comments can shed some light:

- Some users noted that the context in which this kind of events occur can greatly affect the relevance evaluation (for instance, a user wrote s/he does not think that a person can be considered agent of her/his birth, but s/he gave 5 to *hasAgent* because in the birth of an organization the founders are agents; also in a person death usually the dead person is not an agent, but in a suicide this can be the case).
- Some comments report difficulties in understanding the meaning of some properties in relation with this kind of events (for instance, in a person birth, is the new-born a participant? When a company hires somebody, which is the role played by the hired person? Is s/he a patient, a beneficiary, or both?).
- In the final general comment, several users wrote that it is difficult to link the proposed properties to this kind of events.

These comments highlighting the context-dependency of the relevance evaluation and the difficulties in interpreting the meaning of (some) properties in relation with life events, could -- at least partially -- explain the high disagreement.

Intuitively physical confrontations (and, in particular, street clashes, police charges, killings, woundings, etc.) and protest actions (like strikes or protest marches) seem to have a different "semantic nature" with respect to life events (like births and deaths, or marriages) which are often related to formal acts (e.g., marriage, hiring and dismissal, role assumption, etc.) and seem to have a very complex semantic characterization. This could be the reason why some of the properties with a high score for events in Groups 1 and 2 clearly sound less important or even a little odd for events in Group 3: for example, properties like *hasAgent* or *hasPatient*, but also *hasAdvantaged/Damaged*, or *hasPurpose*, seem to have a marginal role in person birth or death, or in a marriage. These considerations suggest us that further deeper ontological analysis of life events should be addressed in our future work (see Section 6).

Before describing the strategy we propose for assigning each property a correct score on the basis of the involved event typology, some extra heuristics are worth to be mentioned, which will contribute to produce the final properties ranking to be used for generating the user interface (see Section 5.4).

In order to preserve the semantic homogeneity, we decided to keep together "related" properties such as all properties referring to time, or location, as well as property "pairs" like *hasAdvantaged/hasDamaged*. Moreover, given that the literature about event modeling (see Section 2 and 3) agrees on considering generic participation, as well as space and time, as fundamental dimensions to model (historical) events, we decided to keep *hasParticipant*, together with all properties referring to time and all properties referring to location, in first position. Finally, we decided to create an ad hoc last position (independently from the results of the

user study) for the generic property *inRelationWith*, since it represents a sort of "backup": if the user wants to state that there is a relation between the event s/he is describing and an entity or another event, but s/he is not able to say which kind of relation (even not the relation of "participation in the event"), s/he can choose *inRelationWith*.
The final result is the matrix shown in Table 7a (Appendix A.2), where the relevance scores (*rs*) obtained for <*property*, *event-class*> pairs have been arranged into four categories:

(A) *Essential* properties (rs >= 4).
(B) *Important* properties (3 <= rs < 4).
(C) *Useful* properties (2 < rs <=3).
(D) Other properties (rs < 2).

In the following section we will describe how these categories are used to generate the user interface.

5.4 The Resulting User Interface

We used the matrix shown in Table 7a (Appendix A.2) to design a new user interface, taking into account the different categories expressing property relevance. The wireframe of such user interface is shown in Figure 8.
The default user interface proposes the user a "guided path" through properties, based on the four relevance categories. However, the user can always access a "flat" form listing all available properties, without taking relevance categories into account (*all properties* link in the wireframe). The flat form is basically the one shown in Figure 5 (see Section 4): properties are displayed in alphabetical order, but two simple heuristics are applied here too: "related" properties (e.g., *hasTimespan*, *hasInitialTime*, etc.) are listed together and *inRelationWith* is left at the end of the list (see Section 5.3).
The guided path, shown in Figure 8, is organized in tabs: the first (default) tab contains *essential properties*, as far as an event (and, in general, an entity) of the selected type is concerned (a strike in Figure 8). The user is invited (but not forced) to start filling these properties, and in particular is invited to first search for a suitable filler in the Semantic KB (*search the KB* link in the figure). Then, she can switch to the second tab, containing *important properties*, and then to *useful properties* (third tab), to conclude with the remaining properties (tab labeled *other* in the figure). It is important to stress that the user is not forced neither to follow the proposed tab sequence, nor to fill any property: the guided path is simply a suggestion in order to avoid presenting the user a long, clumsy, possibly confusing flat form.
Another strategy supporting this goal is the link enabling the user to access the time-related and the location-related properties: since in HERO there are a number of different properties that can be used to specify time (and the same is true for location), these properties are shown only upon request (*other time/location-related properties* link in the figure).

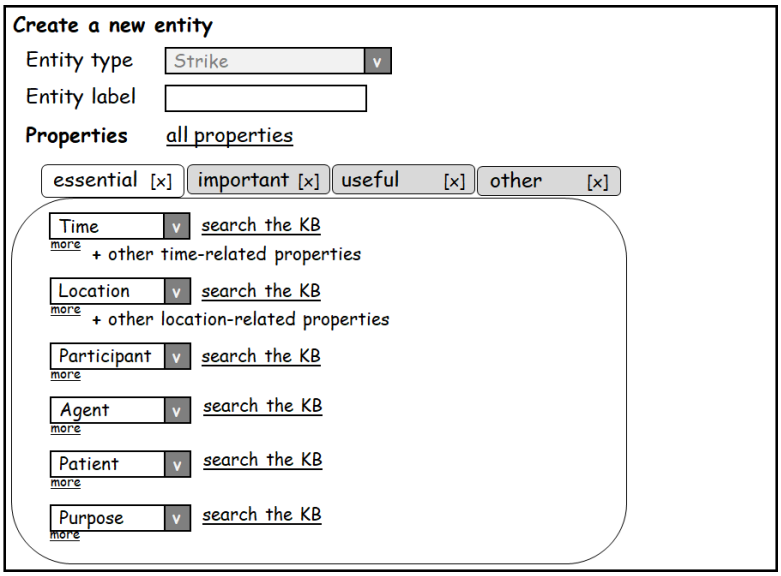

Figure 8: wireframe of the new user interface enabling users to characterize an entity (typically an event) through its properties.

We can now come back to the initial **research question** to state that, starting from the axioms characterizing events and their properties/relations in HERO, the following combined solution supports the generation of a user interface that enables users to characterize events exploiting the conceptual vocabulary provided by HERO:

- A configuration tool enabling administrators to hide classes and properties "unsuited" for human annotators (Figure 7).
- The output of the algorithm described in Section 5.1, that combines heuristics and reasoning to extract from HERO all and only the properties that are compatible with a given event type.
- The results of a user study enabling us to assign to each pair <*property*, *event-class*> one of four relevance categories; such assignments, in turn, supported the design of a new user interface (Figure 8) for the ontology-driven semantic characterization of events in the PRiSMHA platform.

As stated in Section 1, the solution to the lack of effective management and access tools for historical archives can only be faced through hybrid strategies, that integrate automatic approaches to content mining with user-generated content. The results presented in this paper represent a contribution in this direction, since they provide a solution for building a user interface that supports users in creating semantic meta-data based on a rich computational ontology.

6 Conclusions and Future Work

In this paper, we presented our answer to the challenge represented by the ontology-driven user interface in the PRiSMA project. In particular, such an interface enables users to provide full-fledged formal semantic representations of the events narrated in documents from historical archives. Many aspects have been pointed out that deserve further attention.

As far as the ontology is concerned -- as emerged from the discussion of the user study results -- we will revise the actual definition of some properties in HERO (those that revealed difficulties in catching the intended meaning, i.e., *hasExperiencer*, *hasTheme*, *hasProspectiveParticipant*, and the distinction between *hasForce* and *hasCausalFactor/isCausalFactorOf*), as well as the ontological analysis of life events.

We will complete the matrix providing the categorization of properties into relevance classes (Table 7a - Appendix A.2) with the missing HERO classes. Moreover, relevance scores for <*property*, *event-class*> pairs

could be learnt on the basis of the users selecting properties to characterize the different event types. In particular, the matrix provided by the user study can be used to solve the *cold start problem* (i.e., categorizing properties into relevance classes when data about system usage are not available yet); as long as the system collects data about user selections, the initial relevance scores in the matrix could be gradually integrated with those calculated on the basis of users selections, that become more and more statistically relevant as far as users interact with the system.

As regards the user interface, we are planning a user evaluation of the proposed user interface (Figure 8).

Finally, as mentioned in Section 1, we are working on the integration of Named Entity Recognition and Event Mining techniques to provide users of the PRiSMHA platform with suggestions during the annotation process.

ACKNOWLEDGMENTS

This work has been supported by Compagnia di San Paolo Foundation and Università di Torino within the PRiSMHA project (CSTO168023). Thanks to all PRiSMHA collaborators, and special thanks to Rossana Damiano, for her valuable support in the discussion of the issues presented in this paper.

REFERENCES

Agosti, M., Conlan, O., Ferro, N., Hampson, C., and Munnelly, G. (2013). "Interacting with Digital Cultural Heritage Collections via Annotations: The CULTURA Approach". Proceedings of the ACM Symposium on Document Engineering (DocEng'13), Florence, Italy, 13-22.

van den Akker, C., Aroyo, L., Cybulska, A., van Erp, M., Gorgels, P., Hollink, L., Jager, C., Legêne, S., van der Meij, L., Oomen, J., van Ossenbruggen, J., Schreiber, G., Segers, R., Vossen, P., and Wielinga, B. (2010). "Historical Event-based Access to Museum Collections". Applied Artificial Intelligence, 25.

Allen, J. F. (1983). "Maintaining Knowledge about Temporal Intervals". Communications of the ACM, 26(11): 832-843.

Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). "Web Services - Concepts, Architectures and Applications". Springer.

Ashenfelder, M. (2015). "Cultural Institutions Embrace Crowdsourcing". blogs.loc.gov/digitalpreservation/2015/09/cultural-institutions-embrace-crowdsourcing.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F. (Eds.) (2010). "The Description Logic Handbook", 2nd Ed., Cambridge University Press.

de Boer, V. Oomen, J., Inel, O., Aroyo, L., van Staveren, E., Helmich,W., and de Beurs, D. (2015). "DIVE into the Event-Based Browsing of Linked Historical Media". Journal of Web Semantics, 35(3): 152-158.

Bonacchi, C., Bevan, A., Pett, D., Keinan-Schoonbaert, A., Sparks, R., Wexler, J., and Wilkin, N. (2014). "Crowd-sourced Archaeological Research: The MicroPasts Project. Archaeology International", 17: 61-68.

Bonacchi, C., Bevan, A., Keinan-Schoonbaert, A., Pett, D., and Wexler, J. (2019). "Participation in Heritage Crowdsourcing". Museum Management and Curatorship, 34 (2): 166-182.

Borgo, S., and Masolo, C. (2009) "Foundational choices in dolce", in Staab, S., and Studer, R. (Eds.) "Handbook on Ontologies", 2nd Ed., Springer, 361-381.

Boschetti, F., Cimino, A., Dell'Orletta, F., Lebani, G. E., Passaro, L., Picchi, P., Venturi, G., Montemagni, S., and Lenci, A. (2014). "Computational Analysis of Historical Documents: An Application to Italian War Bulletins in World War I and II". Proceedings of Language Resources and Evaluation Conference, Reykjavik, Iceland.

Calvanese, D., Mosca, A., Remesal, J., Rezk, M., and Rull, G. A. (2015). "'Historical Case' of Ontology-Based Data Access". Proceedings of Digital Heritage, Granada, Spain, 291-298.

Carducci, G., Leontino, M., Radicioni, D. P., Bonino, G., Pasini, E., and Tripodi, P. (2019). "Semantically Aware Text Categorisation for Metadata Annotation". Proceedings of the Italian Research Conference on Digital Libraries, Pisa, Italy, 315-330.

Daquino M., Mambelli F., Peroni S., Tomasi F., and Vitali F. (2016). "Enhancing Semantic Expressivity in the Cultural Heritage Domain: Exposing the Zeri Photo Archive as Linked Open Data". Journal on Computing and Cultural Heritage, 10(4).

Doerr, M. (2003) "The cidoc conceptual reference model: an ontological approach to semantic interoperability of metadata". AI Magazine, 24(3): 75-92.

Dragoni, M., Tonelli, S., and Moretti, G. (2016). “A knowledge management architecture for digital cultural heritage”. ACM Transactions on Applied Perception, 1(1).

Europeana (2016). “Definition of the Europeana Data Model v.5.2.7”. pro.europeana.eu/files/Europeana_Professional/Share_your_data/Technical_requirements/EDM_Documentation/EDM_Definition_v5.2.7_042016.pdf

Foley, J., Kwan, P., and Welch, M. (2017). “A web-based infrastructure for the assisted annotation of heritage collections”. ACM J. on Computing and Cultural Heritage, 10(3).

Franconi, E., Guagliardo, P., and Trevisan, M. (2010). “An intelligent query interface based on ontology navigation”. Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web, Hong Kong, China.

Gangemi, A., and Mika, P. (2003). “Understanding the semantic web through descriptions and situations”. OTM Confederated International Conferences on the Move to Meaningful Internet Systems, Catania, Italy, 689-706.

Giretzlehner M, Girardi D, and Arthofer K. (2012). “Ontology-guided data acquisition and analysis: Using ontologies for advanced statistical analysis”. Proceedings of the International Conference on Data Analytics, Barcelona, Spain.

Gonçalves, R. S., Tu, S. W., Nyulas, C. I., Tierney, M. J., and Musen, M. A. (2017). “An ontology-driven tool for structured data acquisition using Web forms”. Journal of Biomedical Semantics, 8:26.

Goy, A., Accornero, C., Astrologo, D., Colla, D., D'Ambrosio, M., Damiano, R., Leontino, M., Lieto, A., Loreto, F., Magro, D., Mensa, E., Montanaro, A., Mosca, V., Musso, S., Radicioni, D. P., and Re, C. (2019a). “Fruitful synergies between computer science, historical studies and archives: the experience in the PRiSMHA project”. Proceedings of the Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Vol. 3: KMIS, 225-230.

Goy, A., Damiano, R., Loreto, F., Magro, D., Musso, S., Radicioni, D., Accornero, C., Colla, D., Lieto, A., Mensa, E., Rovera, M., Astrologo, D., Boniolo, B., and D’ambrosio, M. (2017). ‘PRiSMHA (Providing Rich Semantic Metadata for Historical Archives’. Proceedings of the Contextual Representation of Objects and Events in Language, Bolzano, Italy.

Goy, A., and Magro, D. (2019). “Collections revisited from the perspective of historical testimonies”. *Int. Journal of Metadata, Semantics and Ontologies*, 13(4): 300-316 .

Goy, A., Magro, D., and Baldo, A. (2019b). “A Semantic Web Approach to Enable a Smart Route to Historical Archives”. Journal of Web Engineering, 18(4-6): 287-318.

Goy, A., Magro, D., and Rovera, M. (2015). “Ontologies and historical archives: a way to tell new stories”. Applied Ontology, 10(3/4): 331-338.

Goy, A., Magro, D., and Rovera, M. (2018). “On the Role of Thematic Roles in a Historical Event Ontology”. Applied Ontology, 13: 19-39.

Gruber, T. R. (1995). “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”. International Journal Human-Computer Studies, 43 (5-6): 907-928.

Gupta, M., Li, R., Yin, Z., and Han, J. (2010). “Survey on social tagging techniques”. ACM SIGKDD Explorations Newsletter 12(1): 58-72.

van Hage, W.R., Malaisé, V., Segers, R., Hollink, L., and Schreiber, G. (2011). “Design and use of the Simple Event Model (SEM)”. Journal of Web Semantics, 9(2): 128-136.

Heath, T., and Bizer, C. (2011). “Linked Data: Evolving the Web into a Global Data Space”. Morgan & Claypool.

Hogenboom, F., Frasincar, F., Kaymak, U., and De Jong, F. (2011). “An Overview of Event Extraction from Text”. In Proceedings of the Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web, Bonn, Germany, 48-57.

Horridge M, Brandt S, Parsia B, qnd Rector A. (2014). “A domain specific ontology authoring environment for a clinical documentation system”. Proceedings of the IEEE International Symposium on Computer-Based Medical Systems, New York, USA.

van Hooland, S., Méndez Rodríguez, E., and Boydens, I. (2011). “Between Commodification and Engagement: On the Double-Edged Impact of User-Generated Metadata within the Cultural Heritage Sector”. Library Trends, 59(4): 707-720.

Hyvönen, E., Lindquist, T., Törnroos, J., and MÄkelä, E. (2012). “History on the semantic web as linked data-an event gazetteer and timeline for the world war I”, Proceedings of CIDOC Enriching Cultural Heritage, Helsinki, Finland.

Isaac A. (Ed.) (2013). “Europeana Data Model Primer”, Creative Commons Licence.

Katifori, A., Nikolaou, C., Platakis, M., Ioannidis, Y., Tympas, A., Koubarakis, M., Sarris, N., Tountopoulos, V., Tzoannos, E., Bykau, S., Kiyavitskaya, N., Tsinaraki, C., and Velegrakis, Y. (2011). "The Papyrus Digital Library: Discovering History in the News", Proceedings of the International Conference on Theory and Practice of Digital Libraries (TPDL), LNCS 6966, 465-468.

Kazakov, Y., (2008). "RIQ and SROIQ Are Harder than SHOIQ". Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, Sydney, Australia, 274-284

King, L., Stark, J.F., and Cooke, P. (2016). "Experiencing the Digital World: The Cultural Value of Digital Engagement with Heritage", Heritage & Society, 9(1): 76-101.

Koukopoulos, Z., and Koukopoulos, D. (2019). "Evaluating the Usability and the Personal and Social Acceptance of a Participatory Digital Platform for Cultural Heritage". Heritage 2019, 2(1): 1-26.

Le Boeuf, P., Doerr, M., Ore, C.E., and Stead, S. (2015). "Definition of the CIDOC Conceptual Reference Model" ICOM/CIDOC CRM Special Interest Group. new.cidoccrm. org/Version/version-6.2.2.

Meroño-Peñuela, A. Ashkpour, A., van Erp, M., Mandemakers, K., Breure, L., Scharnhorst, A., Schlobach, S., and van Harmelen, F. (2015). "Semantic Technologies for Historical Research: A Survey". Semantic Web Journal, 6(6): 539-564.

Oomen, J., and Belice, L. (2012). "Sharing cultural heritage the linked open data way: why you should sign up". Proceedings of Museums and the Web Conference, san Diego, USA.

Marchetti, A., Tesconi, M., Ronzano, F., Rosella, M,. and Minutoli, S. (2007). "Semkey: A semantic collaborative tagging system". Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, Calgary, Canada, 8-12.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). "WonderWeb deliverable D18". ISTC-CNR, 2003. Technical Report.

Motta, E., Buckingham Shum, S., and Domingue, J. (2000). "Ontology-driven document enrichment: principles, tools and applications". International Journal of Human-Computer Studies, 52(6): 1071-1109.

Nadeau, D., and Sekine, S. (2007). "A survey of named entity recognition and classification". Lingvisticae Investigationes, 30(1): 3-26.

Nanni, F., Ponzetto, S.P., and Dietz, L. (2017). "Building entity-centric event collections". Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, Toronto, Canada.

Noordegraaf, J., Bartholomew, A., Eveleigh, A. (2014). "Modeling crowdsourcing for cultural heritage". Proceedings of Museums and the Web, Baltimore, USA.

Palmer, M., Gildea, D., and Xue, N. (2010). "Semantic role labeling". Synthesis Lectures on Human Language Technologies, 3(1): 1-103.

Parsia, B., Matentzoglu, N., Gonçalves, R.S., Glimm, B., Steigmiller, A. (2017). "The OWL Reasoner Evaluation (ORE) 2015 Competition Report", Journal of Automated Reasoning, 59(4): 455-482.

Paulheim, H., and Probst, F. (2010). "Ontology-Enhanced User Interfaces: A Survey". Int. Journal of Semantic Web and Information Systems, 6(2).

Raymond, Y., and Abdallah, S. (2007). "The event ontology". Proceedings of the Events and Stories in the News Workshop, Vancouver, Canada, 87-97.

Rector A. (2013). "Axioms & templates: Distinctions & transformations amongst ontologies, frames & information models". Proceedings of the Int. Conference on Knowledge Capture, Banff, Canada.

Ridge, M. (2013). "From Tagging to Theorizing: Deepening Engagement with Cultural Heritage through Crowdsourcing". The Museum Journal, 56(4): 435-450.

Sharp, H., Preece, J., and Rogers, Y. (2019). "Interaction Design: Beyond Human-Computer Interaction", 5th Ed., Wiley.

Shaw, R.,Troncy, R., Hardman, L. (2009). "LODE: Linking Open Descriptions of Events". Proceedings of the Asian Conference on The Semantic Web, Shanghai, China, 153-167.

Scherp, A., Franz, T., Saathoff, C., and Staab, F. (2009). "F - a model of events based on the foundational ontology dolce+dnsultralite". Proceedings of the International Conference on Knowledge Capture, Banff, Canada, 137-144.

SHACL (2017). "Shapes Constraint Language (SHACL)". W3C. www.w3.org/TR/shacl/

Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L., and Dann, J. (2018). "Automated extraction of semantic legal metadata using natural language processing". Proceedings *of the Int. Requirements* Engineering Conference, Banff, Canada, 124-135.

Soylu, A., Giese, M., Jimenez-Ruiz, E., Kharlamov, E., Zheleznyakov, D., and Horrocks, I. (2017). "Ontology-based end-user visual query formulation: Why, what, who, how, and which?". Universal Access Information Society, 16: 435-467.

Sprugnoli, R,. and Tonelli, S. (2017). "One, no one and one hundred thousand events: defining and processing events in an inter-disciplinary perspective". Natural Language Engineering, 23(4): 485-506.
Steigmiller, A., Liebig, T., Glimm, B. (2014). "Konclude: System Description". Journal of Web Semantics, 27(1): 78-85.
Terras, M. (2016). "Crowdsourcing in the Digital Humanities". In Schreibman, S., Siemens, R., and Unsworth, J. (Eds.) "A New Companion to Digital Humanities", Wiley-Blackwell, 420-439.
Tunkelang, D., Marchionini, G. (2009). "Faceted Search. Retrieval, and Services". Morgan and Claypool Publishers.
Vamvakas, G., Gatos, B., and Perantonis, S. J. (2010). "Handwritten character recognition through two-stage foreground sub-sampling". Pattern Recognition, 43(8): 2807-2816.
Visser, J., and Richardson, J. (2013). "Digital Engagement in Culture, Heritage and the Arts". Creative Commons Attribution-Share Alike license.
Zarri, G.P. (2015). "A structured and in-depth representation of the semantic content of elementary and complex events". Int. Journal of Metadata, Semantics and Ontologies, 10(1): 12-27.

A APPENDICES

A.1 Mean and Standard Deviation Values

Table 1a: Mean values for Group 1. Properties are classified according to the relevance scores obtained (exceptions to the proposed classification are marked in red).

MEAN	Physical Confront.	Beating	Street Clash	Police Charge	Firing	Killing People	Wound.
value greater than 4 for all classes							
hasAgent	4,4	4,6	4,4	4,3	4,9	4,7	4,3
hasCausalFactor/ isCausalFactorOf	4,1	4,1	4,2	4,2	4,3	4,3	4,2
hasDamaged	4,2	4,1	4	4,3	4,3	4,3	4,2
hasInitialTime	4,1	4	4,2	4,4	4,3	4,3	4,1
hasLocation	4,3	4,4	4,7	4,6	4,7	4,7	4,6
hasParticipant	4,6	4,4	4,3	4,2	4,4	4,9	4,3
hasPatient	4,3	4,4	4,2	4,6	4,7	4,7	4,4
hasPurpose	4	4,1	4,6	4,2	4,2	4,4	4,2
hasTimespan	4	4,2	4,2	4,3	4,3	4,3	4
hasFinalTime	4	4,2	4,2	4,4	4	4,3	3,9
value greater than 4 for some classes, but between 3 and 4 for other classes							
hasBeginningEvent/ isBeginningEventOf	3,2	3,3	4,1	3,7	4	4,1	3,7
hasFinalLocation	3,8	3,7	4,1	4,1	4,1	3,9	3,7
hasForce	3,1	3,1	3,4	3,7	4	4,2	3,7
hasInitialLocation	3,6	3,4	4,1	4,1	3,8	3,7	3,6
hasInstrument	3,8	4,1	3,4	3,7	4	4,7	4,4
hasOpponent	3,7	4	4,1	4,2	4,1	4,3	3,9
hasResult	3,4	3,6	4,6	3,7	4,6	4,7	4,3
hasRetaliation/ isRetaliationFor	3,8	3,9	4	3,9	4,1	4,1	4
Influences/isInfluencedBy	3,8	3,8	4	3,6	4,3	4,3	3,9
value between 3 and 4 for all classes							
hasActionTopic	3,9	4	4	3,8	3,7	3,8	3,7
hasAdvantaged	3,3	3,3	3,6	3,7	3,7	3,6	3,2
hasEndingEvent/ isEndingEventOf	3,1	3,1	3,4	3,4	3,4	3,6	3,2
hasExperiencer	3,3	3,4	3,1	3,1	3,3	3,7	3,6
hasSubEvent/isSubEventOf	3,9	3,8	3,8	3,7	4	4	4
hasSupporter	3,6	3,6	3,8	3,4	3,7	4	3,4
inRelationWith	3,4	3,4	3,4	3,6	3,6	3,6	3,4

value between 3 and 4 for some classes, but between 2 and 3 for other classes							
hasIntermediateLocation	2,8	2,7	3,3	3,3	2,8	2,8	2,7
hasProspectiveParticipant	2,2	2,3	3,1	3	3,1	3,2	2,7
value between 2 and 3 for all classes							
hasTheme	2,3	2,3	2,6	2,6	2,6	2,6	2,4

Table 2a: Standard deviation values for Group 1. Properties are classified according to the scores obtained (exceptions to the proposed classification are marked in red). Given that the maximum st. dev. for each couple <property, class> is 2.1, we can consider a low score -- i.e., a fundamental agreement -- values lower than 1.0, and a high score -- i.e., a substantial disagreement -- values greater than 1.0.

ST. DEV.	Physical Confront.	Beating	Street Clash	Police Charge	Firing	Killing People	Wound.
low score (fundamental agreement) for all classes							
hasAgent	0,7	0,7	0,5	0,7	0,3	0,7	0,7
hasLocation	0,7	0,7	0,5	0,5	0,7	0,7	0,5
hasParticipant	0,5	0,7	0,9	1,0	0,5	0,3	0,7
hasPatient	0,7	0,7	1,0	0,5	0,5	0,7	0,7
score only slightly lower than 1.0 (moderate agreement) for all classes							
hasCausalFactor/ isCausalFactorOf	0,8	0,8	0,8	0,8	0,7	0,9	0,8
hasDamaged	0,8	0,9	0,9	0,7	0,7	0,7	0,8
hasInitialTime	0,8	1,1	0,7	0,7	0,9	0,9	0,9
hasOpponent	0,9	1,0	0,9	0,8	0,9	0,9	0,9
hasPurpose	0,9	0,8	0,7	0,8	0,8	0,7	0,8
hasSupporter	0,9	0,9	1,0	0,9	1,0	1,0	0,9
hasAdvantaged	0,7	0,7	0,7	0,9	0,9	1,1	1,0
hasTimespan	1,1	0,8	0,8	0,9	0,9	0,9	1,0
Influences/isInfluencedBy	1,0	1,0	0,7	1,1	0,9	0,9	0,9
different scores depending on the event class (typology)							
hasFinalTime	0,9	0,8	0,8	0,7	1,1	0,9	1,2
hasForce	0,9	0,9	1,1	1,1	1,1	0,7	0,9
hasInstrument	1,0	1,1	1,1	1,0	0,9	0,7	0,7
hasResult	1,4	1,4	0,5	1,4	0,7	0,5	0,9
score only slightly greater than 1.0 (moderate disagreement) for all classes							
hasFinalLocation	1,1	1,0	1,1	1,1	1,1	1,1	1,0
hasInitialLocation	1,2	1,1	1,3	1,3	1,2	1,2	1,2
hasSubEvent/isSubEventOf	1,2	1,2	1,0	1,0	1,0	1,0	1,0
hasTheme	1,0	1,0	1,1	1,1	1,2	1,2	1,1
high score (substantial disagreement) for all classes							
hasActionTopic	1,4	1,3	1,3	1,5	1,4	1,5	1,4
hasEndingEvent/ isEndingEventOf	1,3	1,4	1,3	1,4	1,4	1,3	1,3
hasExperiencer	1,3	1,3	1,2	1,2	1,2	1,4	1,4
hasIntermediateLocation	1,5	1,3	1,5	1,5	1,4	1,4	1,3
hasProspectiveParticipant	1,0	1,2	1,3	1,5	1,7	1,6	1,5
hasRetaliation/isRetaliationFor	1,5	1,4	1,4	1,5	1,4	1,4	1,3
inRelationWith	1,2	1,4	1,2	1,2	1,4	1,4	1,4
hasBeginningEvent/ isBeginningEventOf	1,4	1,4	0,9	1,3	1,2	1,2	1,3

Table 3a: Mean values for Group 2. Properties are classified according to the relevance scores obtained (exceptions to the proposed classification are marked in red).

MEAN	Protest Action	Demon-stration	Strike	Picketing	Protest March	PlaceOc-cupation	SitIn
value greater than 4 for all classes							
hasActionTopic	4,6	4,6	4,6	3,9	4,4	4	4,4
hasAgent	4,8	4,6	4,7	4,6	4,3	4,7	4,7
hasCausalFactor/ isCausalFactorOf	4,3	4,2	4,4	4,1	4,4	4,4	4,2
hasDamaged	4,6	4,4	4,3	4,7	4,1	4,4	4,6
hasFinalTime	4	4	4,2	4	4	4,1	4
hasInitialTime	4,1	4	4,3	4	3,9	4,3	4
hasLocation	4,3	4,2	4,4	4,4	4,4	4,7	4,4
hasOpponent	4,1	4,1	4,1	4,1	4,1	4,1	4,2
hasParticipant	4,4	4,1	4,2	4	4,1	4,1	4,2
hasPatient	4,4	4,6	4,6	4,4	4,4	4,3	4,6
hasPurpose	4,7	4,7	4,7	4,6	4,8	4,7	4,8
hasResult	4,6	4,3	4,4	4,1	4,2	4,1	4,6
hasSupporter	4	4	4,1	4	4,3	3,9	4,2
hasTimespan	4	4	4,2	4	4	4,1	4
value greater than 4 for some classes, but between 3 and 4 for other classes							
hasAdvantaged	4,1	3,8	4,2	3,7	3,9	3,4	3,8
hasBeginningEvent/ isBeginningEventOf	3,9	3,6	4,2	3,8	3,8	4,1	3,9
hasEndingEvent/ isEndingEventOf	3,8	4	4	3,8	4	3,8	4,1
hasExperiencer	4,1	3,8	3,6	4,2	3,3	4,1	4,3
hasFinalLocation	3,9	4,2	3,4	3,9	3,8	4,1	3,7
hasInitialLocation	3,9	3,9	3,9	4	3,9	4,3	4,1
hasInstrument	4,2	3,8	3,6	3,4	3,6	3,7	3,9
Influences/isInfluencedBy	4	3,9	4,3	3,8	4,1	4	4,1
value between 3 and 4 for all classes							
hasForce	3,9	3,9	4	3,8	3,9	4	4
hasProspectiveParticipant	3,4	3,8	3,8	3,7	3,3	3,4	3,6
hasRetaliation/ isRetaliationFor	3,9	3,9	4	4	4,1	4	3,9
hasSubEvent/isSubEventOf	3,7	3,6	3,6	3,6	3,4	3,2	3,6
hasTheme	3,3	3,2	3,3	3,3	3,3	3,3	3,4
inRelationWith	3,1	3	3,2	3,2	3	3,3	3,2
value between 3 and 4 for some classes, but between 2 and 3 for other classes							
hasIntermediateLocation	3,3	3,2	3	3,1	3,1	2,8	2,6

Table 4a: Standard deviation values for Group 2. Properties are classified according to the relevance scores obtained (exceptions to the proposed classification are marked in red). We can consider a low score -- i.e., a fundamental agreement -- values lower than 1.0, and a high score -- i.e., a substantial disagreement -- values greater than 1.0.

ST. DEV.	Protest Action	Demon-stration	Strike	Picketing	Protest March	Place Oc-cupation	SitIn
low score (fundamental agreement) for all classes							

hasAgent	0,4	0,7	0,5	0,7	0,7	0,5	0,5
hasDamaged	0,7	0,9	0,9	0,5	0,9	0,7	0,7
hasFinalTime	0,7	0,7	0,7	0,7	0,7	0,8	0,7
hasInitialTime	0,8	0,7	0,7	0,7	0,8	0,5	0,7
hasInstrument	0,7	0,8	0,7	0,9	0,7	0,7	0,6
hasPatient	0,7	0,7	0,7	0,9	0,7	0,9	0,7
hasPurpose	0,7	0,7	0,7	0,7	0,4	0,5	0,4
inRelationWith	0,6	0,5	0,7	0,7	0,7	0,5	0,7
score only slightly lower than 1.0 (moderate agreement) for all classes							
hasActionTopic	0,7	0,7	0,9	0,9	0,7	0,9	0,9
hasAdvantaged	0,9	1,0	0,7	1,0	0,9	1,0	1,1
hasCausalFactor/ isCausalFactorOf	0,9	0,8	0,9	0,9	0,7	0,7	1,0
hasEndingEvent/ isEndingEventOf	1,2	1,0	0,7	0,8	0,9	1,0	0,8
hasLocation	1,0	1,0	1,0	0,7	0,7	0,5	0,7
hasParticipant	0,7	0,8	1,0	0,9	0,6	0,9	0,8
hasSubEvent/isSubEventOf	1,0	0,9	1,1	1,0	0,9	1,1	0,9
hasSupporter	1,0	0,7	0,6	0,9	0,7	0,9	0,7
hasTheme	0,9	1,0	1,1	1,1	1,0	0,9	1,0
hasTimespan	0,9	0,9	0,8	0,9	0,9	0,8	0,7
different scores depending on the event class (typology)							
hasExperiencer	0,9	1,0	1,2	1,0	1,2	1,1	0,7
hasFinalLocation	0,8	0,7	1,2	0,9	1,2	1,1	0,7
hasOpponent	1,3	0,6	0,8	0,8	1,1	0,8	0,7
hasResult	0,7	0,9	1,0	1,1	1,4	1,4	0,7
score only slightly greater than 1.0 (moderate disagreement) for all classes							
hasBeginningEvent/ isBeginningEventOf	1,2	1,1	1,1	1,1	1,1	1,1	1,3
hasForce	1,5	1,1	1,0	1,2	1,3	1,0	1,1
hasInitialLocation	1,1	0,9	1,1	1,1	1,1	0,7	0,8
hasIntermediateLocation	1,2	1,2	1,0	1,1	1,3	1,1	1,0
Influences/isInfluencedBy	1,1	1,1	0,9	1,1	0,9	1,1	0,9
high score (substantial disagreement) for all classes							
hasProspectiveParticipant	1,4	1,4	1,2	1,4	1,1	1,3	1,3
hasRetaliation/isRetaliationFor	1,5	1,3	1,3	1,3	1,3	1,2	1,5

Table 5a: Mean values for Group 3. Properties are classified according to the relevance scores obtained (exceptions to the proposed classification are marked in red).

MEAN	Person Birth	Person Death	Marriage	RoleAs-sumption	Hiring	Dismissal	Retiring
value greater than 4 for all classes							
hasCausalFactor/ isCausalFactorOf	4,3	4,6	4	4,6	4	4,7	4,1
hasParticipant	4,1	4,1	4,1	4,3	4,1	4,3	4
hasTimespan	4,2	4,2	4,2	4	4	4	4
hasSpouse			4,7				
hasAssumedRole				4,8			
value greater than 4 for some classes, but between 3 and 4 for other classes							

hasAdvantaged	3,2	3,2	3,1	4,4	4	3,4	2,9
hasAgent	4	3,9	3,7	4,8	4,6	4,9	4,6
hasLocation	3,9	3,8	4,1	4	4,2	3,3	2,8
hasResult	4	4,2	3	4,7	3,7	3,4	3,1
Influences/isInfluencedBy	3,4	3,9	3,2	4,6	4,2	4,6	3,4
value between 3 and 4 for all classes							
hasSubEvent/isSubEventOf	3,1	3,3	3,1	3,8	3,4	4	3,1
inRelationWith	3,8	4	3,2	3,9	3,4	3,9	3,1
value between 3 and 4 for some classes, but between 2 and 3 for other classes							
hasActionTopic	2,7	2,7	2,1	3,7	3,4	3,1	2,4
hasBeginningEvent/ isBeginningEventOf	3,4	3,7	2	3,4	3	3,3	2,6
hasDamaged	2,8	3,8	2,4	3,7	3,1	4,6	3
hasEndingEvent/ isEndingEventOf	3,1	3,6	2,3	3,7	3,2	3,6	3,2
hasExperiencer	3	3,6	2,1	4,3	4	4,2	3,4
hasFinalLocation	3	3,6	2,3	3,8	3,2	3,3	2,9
hasFinalTime	2,4	3,3	2,9	3,9	3,3	3,2	3,2
hasForce	2,9	4,2	2,1	3,3	3	4,1	2,7
hasInitialLocation	3,2	3,3	2,2	3,6	3,6	3,2	2,7
hasInitialTime	3,7	3,7	2,8	3,8	3,7	3,6	3,2
hasInstrument	2,1	3,4	2,1	2,9	3,4	3,7	2,4
hasIntermediateLocation	2,9	3	1,9	3,6	3	2,9	2,8
hasOpponent	2,8	3,3	2,6	4,4	3,9	3,9	2,8
hasPatient	3,2	4	2,6	3,4	3,3	4,4	3,4
hasProspectiveParticipant	2,8	2,9	2,2	4,3	3,6	3,7	2,4
hasPurpose	2,8	3,4	2,4	4,3	3,7	4	3,4
hasRetaliation/isRetaliationFor	2,6	4,1	2,1	3,4	2,7	4,2	2,7
hasSupporter	2,8	3,3	2	4,6	4	4,4	3,2
hasTheme	2,6	2,7	1,7	4,1	4	3,9	3

Table 6a: Standard deviation values for Group 3. Properties are classified according to the relevance scores obtained (exceptions to the proposed classification are marked in red). We can consider a low score -- i.e., a fundamental agreement -- values lower than 1.0, and a high score -- i.e., a substantial disagreement -- values greater than 1.0.

ST. DEV.	Person Birth	Person Death	Marriage	RoleAs-sumption	Hiring	Dismissal	Retiring
low score (fundamental agreement) for all classes							
hasSpouse			0,5				
hasAssumedRole				0,7			
score only slightly lower than 1.0 (moderate agreement) for all classes							
hasAgent	0,9	1,1	1,2	0,4	0,7	0,3	0,5
hasCausalFactor/isCausalFactorOf	0,7	0,7	0,9	0,7	1,3	0,7	0,8
hasParticipant	0,8	0,9	0,6	1,0	0,9	1,0	0,9
different scores depending on the event class (typology)							
hasOpponent	1,6	1,2	1,3	0,5	1,5	1,5	1,3
hasResult	1,5	1,3	1,1	0,5	1,5	1,7	1,7
hasSupporter	1,2	1,1	0,7	0,7	1,4	1,3	1,6
hasTimespan	1,1	1,1	1,0	0,7	0,9	1,1	0,7
score only slightly greater than 1.0 (moderate disagreement) for all classes							
hasLocation	1,1	1,3	1,2	1,0	0,8	0,9	1,2

hasPatient	1,1	1,0	1,4	1,4	1,6	0,7	1,2
hasProspectiveParticipant	1,4	1,4	1,1	1,0	1,4	1,7	1,2
Influences/isInfluencedBy	1,4	1,4	1,4	0,5	1,0	1,0	1,2
inRelationWith	1,6	1,3	1,5	1,3	1,5	1,3	1,4
high score (substantial disagreement) for all classes							
hasActionTopic	1,7	1,5	1,1	1,3	1,3	1,8	1,6
hasAdvantaged	1,6	1,4	1,2	0,7	1,5	1,6	1,6
hasBeginningEvent/ isBeginningEventOf	1,7	1,7	1,1	1,6	1,7	1,9	1,7
hasDamaged	1,3	1,6	1,2	1,7	1,8	1,3	1,5
hasEndingEvent/isEndingEventOf	1,8	1,6	1,5	1,7	1,8	1,9	1,9
hasExperiencer	1,5	1,7	1,5	1,3	1,7	1,4	1,7
hasFinalLocation	1,7	1,8	1,5	1,2	1,5	1,5	1,5
hasFinalTime	1,8	1,9	1,7	1,3	1,7	1,6	1,6
hasForce	1,7	1,6	0,9	1,1	1,3	1,4	1,1
hasInitialLocation	1,6	1,9	1,5	1,2	1,2	1,5	1,4
hasInitialTime	1,7	1,7	1,6	1,3	1,3	1,7	1,5
hasInstrument	1,5	1,9	1,2	1,9	1,6	1,7	1,4
hasIntermediateLocation	1,6	1,7	1,2	1,1	1,2	1,4	1,3
hasPurpose	1,7	1,7	1,5	1,3	1,4	1,5	1,7
hasRetaliation/isRetaliationFor	1,7	1,1	1,5	1,6	1,8	0,8	1,5
hasSubEvent/isSubEventOf	1,7	1,7	1,4	1,3	1,2	1,2	1,5
hasTheme	1,6	1,7	0,9	1,4	1,3	1,7	1,7

A.2 Relevance Categories

Table 7a: The matrix providing the relevance category for <*property*, *event-class*> pairs.

event class → property ↓	Beating	Demon-stration	Dismissal	Firing	Hiring	Killing People	Marriage	Person Birth	Person Death	Physical Confront.	Picketing	Place Occupat.	Police Charge	Protest Action	Protest March	Retiring	Role As-sumption	Sit In	Street Clash	Strike	Wound
hasActionTopic	A	A	B	B	B	B	C	C	C	B	B	A	B	A	A	C	B	A	A	A	B
hasAdvantaged	A	A	B	A	A	A	B	B	B	A	A	A	A	A	A	C	A	A	A	A	A
hasAgent	A	A	A	A	A	A	B	A	B	A	A	A	A	A	A	A	A	A	A	A	A
hasAssumedRole	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	A	/	/	/	/
hasBeginningEvent/isBeginningEventOf	B	B	B	A	B	A	C	B	B	B	B	A	B	B	B	C	B	B	A	A	B
hasCausalFactor/isCausalFactorOf	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
hasDamaged	A	A	B	A		A	B	B	B	A	A	A	A	A	A	C	A	A	A	A	A
hasEndingEvent/isEndingEventOf	B	B	B	A	B	A	C	B	B	B	B	A	B	B	B	C	B	B	A	A	B
hasExperiencer	B	B	A	B	A	B	C	B	B	B	A	A	B	A	B	B	A	A	B	B	B
hasFinalLocation	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A

hasFinalTime	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
hasForce	B	B	A	A	B	A	C	B	A	B	B	A	B	B	B	C	B	A	B	A	B	
hasInitialLocation	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
hasInitialTime	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
hasInstrument	A	B		A		A				B	B	B	B	A	B			B	B	B	A	
hasIntermediateLocation	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
hasLocation	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
hasOpponent	A	A	B	A	B	A	C	C	B	B	A	A	A	A	A	C	A	A	A	A	B	
hasParticipant	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
hasPatient	A	A	A	A	B	A	C	B	A	A	A	A	A	A	A	B	B	A	A	A	A	
hasProspectiveParticipant	C	B	B	B	B	B	C	C	C	C	B	B	B	B	B	C	A	B	B	B	C	
hasPurpose	A	A	A	A	B	A	C	C	B	A	A	A	A	A	A	B	A	A	A	A	A	
hasResult	B	A	B	A	B	A	B	A	A	B	A	A	B	A	A	B	A	A	A	A	A	
hasRetaliation/isRetaliationFor	B	B	A	A	C	A	C	C	A	B	A	A	B	B	A	C	B	B	A	A	A	
hasSpouse	/	/	/	/	/	/	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/
hasSubEvent/isSubEventOf	B	B	A	A	B	A	B	B	B	B	B	B	B	B	B	B	B	B	B	B	A	
hasSupporter	A	A	B	A	B	A	C	C	B	B	A	A	A	A	A	C	A	A	A	A	B	
hasTheme	C	B	B	C	A	C	D	C	C	C	B	B	C	B	B	B	A	B	C	B	C	
hasTimespan	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A
Influences/isInfluencedBy	B	B	A	A	A	A	B	B	B	B	B	A	B	A	A	B	A	A	A	A	B	
inRelationWith	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--