A Pipeline Supporting a Smart Access to Historical Documents based on a Rich Semantic Representation of Their Content: A Case Study on Time Expressions

A Pipeline Supporting a Smart Access to Historical Documents based on a Rich Semantic Representation of their Content

A Case Study on Time Expressions *

Alessandro Baldo[1], Anna Goy[1] and Diego Magro[1]
[1] *Dipartimento di Informatica, Università di Torino, C. Svizzera 185, Torino, Italy*
baldoalessandro@protonmail.com, {annamaria.goy, diego.magro}@unito.it

Keywords: Semantic Web, Web-based Intelligent Systems, Ontology, Web Services, Digital Humanities.

Abstract: This work is part of two ongoing projects whose main goal is to demonstrate how semantic technologies can support an effective access to historical archives. In this paper we present a full pipeline, from rough texts up to the final user interface, aimed at creating and exploiting such representations. The pipeline is structured in three modules - handling information extraction, semantic representations, and queries - and offers external applications the possibility of accessing, and thus re-using, the output of each module, by providing a tagged text, a SPARQL endpoint, and a RESTful web service. In the paper, we describe the details of a proof-of-concept implementation of the pipeline architecture that focuses on time expressions. Moreover, we present an example application that exploits the pipeline to enable users to access historical documents by searching and browsing events and time specifications, thus demonstrating the effectiveness of an access to historical texts based on a rich semantic representation of their content.

1 INTRODUCTION

This work is part of two ongoing projects, Harlock'900 (di.unito.it/harlock900) and PRiSMHA (di.unito.it/prismha), based on a collaboration between Università di Torino (Computer Science and Historical Studies departments) and Fondazione Istituto Piemontese A. Gramsci (part of the Polo del '900 foundation: www.polodel900.it), and funded by Università di Torino and Compagnia di San Paolo. Both projects, from slightly different perspectives, aim at the valorization of historical archives and their content through semantic technologies. The main goal is to demonstrate how a rich formal semantic representation of the content of historical documents can provide historians, researchers, journalists, students, and simply interested users with an effective access to the information contained in historical archives. In particular, within PRiSMHA, we are investigating crowdsourcing approaches to overcome the bottleneck represented by the production of rich formal semantic representations of the content of historical documents (Goy et al., 2017).

With respect to this focus, in this paper we present a complementary perspective: a full pipeline, from rough texts up to the final User Interface (UI), aimed at creating and exploiting semantic representations of the content of historical documents. In line with the major trend in the research field, such representations are centered on the notion of *event*, together with its *properties*, i.e., *place*, *time*, and *participants* (people, organizations, collectives) with their *roles* (see Section 2).

The pipeline is structured in three macro-steps, corresponding to its three main modules, namely:

- *Information Extraction module*: Identification/ extraction of descriptions of events and related expressions (referring to time, places, and participants) in texts (see Section 3.2);
- *Semantic module*: Construction of a rich semantic characterization of the identified events (see Section 3.3);

*POSTPRINT VERSION. Cite as: Baldo, A., Goy, A., and Magro, D.. A Pipeline Supporting a Smart Access to Historical Documents based on a Rich Semantic Representation of Their Content: A Case Study on Time Expressions. In Proceedings of the 14th Int. Conf. on Web Information Systems and Technologies (WEBIST 2018), pages 199-206 - ISBN: 978-989-758-324-7.

- *Query Support module*: Definition and implementation of an indexing data structure supporting efficient queries (see Section 3.4).

Moreover, we implemented an example application (see Section 4) that exploits the output of the Query Support module) to enable users to access historical documents by searching and browsing the semantic representation of their content.

The most notable aspect of the pipeline described in this paper is the possibility of accessing, and thus re-using, the output of its main modules: different kinds of applications – e.g., digital libraries, tourist guides, education tools, citizen services – could exploit the knowledge offered by the different mentioned modules, as we will describe in detail in the rest of the paper: after a brief review of major related works (Section 2), in Section 3 we will introduce the pipeline architecture and its three main modules. In Section 4 we will present a web-based application exploiting the pipeline, while Section 5 will conclude the paper.

2 RELATED WORK

As already stated in Section 1, in this paper we will present a pipeline, composed of different modules, mainly handling Information Extraction (IE) from historical texts and generation of formal semantic representation of their content. It is clear, in this perspective, that the related work is huge and spans many research fields, preventing us from providing a complete overview. In this section we therefore survey only the most relevant approaches, without any claim of exhaustiveness.

The core of the projects in which the pipeline is grounded is represented by the ontology-based semantic representation of events narrated within historical documents. The exploitation of Semantic Web technologies within historical research areas has grown in the last decade (Meroño-Peñuela et al., 2015), (Goy et al., 2015), as proven by large projects such as Europeana (www.europeana.eu). Dealing with the formal representation of knowledge about cultural heritage, it is impossible not to mention CIDOC Conceptual Reference Model (www.cidoc-crm.org), an ISO standard that has been used in many projects about cultural heritage – e.g., WarSampo (seco.cs.aalto.fi/projects/sotasampo/en), a project aiming at publishing datasets about the Second World War in Finland as Linked Open Data.

CIDOC-CRM, as well as other ontologies in the same domain, like SEM (van Hage et al., 2011), are centered on the notion of *event*, usually characterized by a set of properties describing "who does what when and where", i.e., time, place and participants of the event itself. The concept of *event* plays a major role also within other projects focusing on the historical domain, such as Agora (van den Akker et al., 2010) and DIVE (de Boer et al., 2015), supporting professional and simply interested users in event-centric browsing of cultural heritage items belonging to different collections. The centrality of the notion of *event* is also demonstrated by the significant research effort that has been devoted to the task of automatically extracting information about events from texts; see, for instance, (Hogenboom et al., 2011), (Cybulska and Vossen, 2011), (Segers et al., 2011), (Sprugnoli and Tonelli, 2016). Moreover, events and their *temporal dimension* play a central role in many approaches aimed at providing an access to cultural heritage collections based on *narratives*, i.e., paths of related events and entities playing relevant roles in them (e.g., famous historical characters); see, for example, Storyscope (Mulholland et al., 2015), among others.

In the pipeline presented in this paper, the input to the Semantic module is provided by the Information Extraction module, aimed at identifying information items within textual documents. With respect to this task, the reference research area is IE, and in particular Named Entity Recognition (NER), with a specific attention to historical texts and to the Italian language. Historical texts represent a peculiar domain, where IE/NER tools show quite low performances compared to other domains (such as news, for example); see, for example (Ehrmann et al., 2016), (Boschetti et al., 2014), (Moretti et al., 2016), (Rovera et al., 2017). In particular, in (Rovera et al., 2017) the authors present the usage of NER tools to extract relevant information from historical texts written in Italian (namely, memories of the Second World War in Italy) and discuss the poor performances of such tools on this specific domain, probably due to the high specificity of the entities to identify: in fact, if state-of-the-art tools have usually good performances on well-known entities (such as "Benito Mussolini"), they often fail with entities that are relevant only in specific historical contexts (e.g., "Nicola Barbato").

3 PIPELINE PROOF-OF-CONCEPT PROTOTYPE

Historical textual sources (biographies, narratives, letters, etc.) abound with descriptions of events and temporal expressions placing these events in time. For this reason, the proof-of-concept implementation of the architecture we are going to present focuses on *time expressions*.

The most interesting aspect of such a prototype is that it enables users and third-party applications to access information about historical events narrated in textual sources by exploiting not only classical calendar-based temporal references, but much more flexible time interval specifications, that can be described thanks to the expressive power of the underlying semantic model (see Section 3.3). From this perspective, the prototype represents a proof of the feasibility and effectiveness of an access to historical texts based on a rich semantic representation of their content.

3.1 Overview of the Architecture

Figure 1 provides an overview of the pipeline architecture.

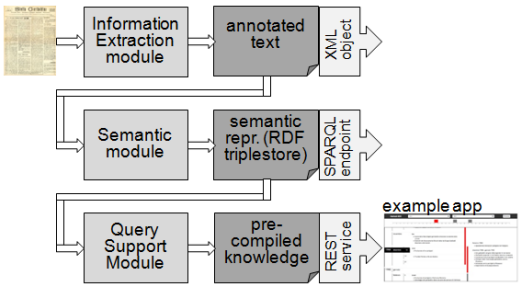

Figure 1: Pipeline architecture.

In the proof-of-concept implementation of the *Information Extraction Module*, to identify temporal expressions from historical texts, we relied on HeidelTime (Strötgen and Gertz, 2013) – probably the most used multilingual, cross-domain, rule-based temporal tagger – that annotates the input text using TimeML (www.timeml.org) (Pustejovsky, 2017). HeidelTime supports the Italian language (Strötgen et al., 2014) (Caselli and Sprugnoli, 2015), and its performance in Italian processing was improved for the participation in the evaluation contest EVENTI-14 (Manfredi et al., 2014), where it achieved good results compared to other systems (Caselli et al., 2014). The identification of events, instead, was based on a manual extraction performed in a previous work (Caserio et al., 2017).

In line with the Semantic Web and Linked Data principles (Heath and Bizer, 2011), the semantic characterization of events and temporal intervals, produced by the *Semantic module*, is stored in an RDF triplestore (based on Apache Jena TDB: jena.apache.org/documentation/tdb); it relies on the HERO ontology (Caserio et al., 2017) and on a declarative (configurable) mapping between TimeML and HERO (see Section 3.3).

As discussed in Section 3.4, since the temporal characterizations contained in the triplestore does not support, at runtime, an efficient retrieval needed to answer flexible temporal queries, the *Query Support module* provides an efficient indexing data structure, enabling temporal range queries.

The proposed pipeline offers third-party applications a set of programming interfaces to access the output of its main modules:

- The output of the Information Extraction Module is a text containing tagged temporal expressions in TimeML format, provided as an *XML object*;
- The semantic representations produced by the Semantic module and stored in the RDF triplestore are accessible through a *SPARQL endpoint* (based on Apache Jena Fuseki: jena.apache.org/documentation/fuseki2);
- The data structure managed by the Query Support module is accessible through a RESTful web service that returns a JSON object (a serialized tree of time intervals IRIs: see Section 3.4).

To test the proof-of-concept prototype we used a selection of texts extracted from two books (in Italian) containing an autobiographical chronological account of events of the "Resistenza" (the Italian struggle against the Fascist regime and the Nazi occupation) in Piemonte (North-West of Italy) during the Second World War: Diena Marisa, *Guerriglia e autogoverno*, Guanda, 1970; Diena Marisa, *Un intenso impegno civile*, Lupieri, 2006.

3.2 Information Extraction Module

The task of temporal tagging can be viewed as a particular Named Entity Recognition task, including three main phases: extraction (i.e., identification of temporal expressions), classification (i.e., association of the temporal expression with a class of an ontology), and normalization (assignment of

the same value to all the time expressions referring to the same temporal entity).

As already stated, the Information Extraction module is based on HeidelTime, that annotates the input text using TimeML. For the scope of this work, the relevant TimeML tag is *TIMEX3* with its four main attributes: *tid* (a per-document identifier); *type* (one out of DATE, TIME, DURATION, or SET); *value* (the normalized value for the marked temporal expression); *mod* (the normalized value for the semantics of modifiers expressions, like – for instance – the value APPROX for the modifier *around* in the expression "around five o'clock").

After annotating the selected texts, we found out some recurring errors that can be classified in two categories: (*a*) errors due to the HeidelTime heuristics; (*b*) errors due to incomplete and/or erroneous rules.

Errors of type *a* are produced by the wrong normalization of *relative* (under-specified) temporal expressions (Strötgen and Gertz, 2013), such as "July" ("luglio"), which require additional contextual information to be correctly normalized. HeidelTime anchors these expressions to the closest temporal expression occurring before the under-specified one. However, this heuristic is not very accurate, as acknowledged also by HeidelTime authors themselves in (Strötgen and Gertz, 2013): for example, if the expression "born in Torino on 13/6/1918" ("nato a Torino il 13/6/1918") precedes the expression "July" ("luglio"), the latter is normalized as "July 1918", even though it is clear from the overall context that it refers to "July 1943". This kind of errors is not influenced by the language of the input document and is deeply rooted in HeidelTime internal heuristics: no modification to the rules could, alone, solve this problem. Therefore, we decided to manually correct these mis-normalizations (which, in our case, occurred in about 50% of the correctly identified temporal expressions of that kind).

Errors of type *b* are related to the identification phase: Temporal expressions like "the morning of June 20th" ("la mattina del 20 giugno") were recognized as two separate expressions instead of a single one (the equivalent English expressions were correctly extracted and normalized). The correct behavior in this case is reported in the Italian TimeML annotation guidelines (Caselli and Sprugnoli 2015). This means that HeidelTime resources for the Italian language were missing the correct rules to handle these expressions: we thus added them to the Italian rules file.

The output of the Information Extraction Module is an XML object containing the original text with TimeML tags identifying temporal expressions.

3.3 Semantic Module

To map temporal expressions identified by the Information Extraction module to RDF triples expressed using the vocabulary provided by the selected ontology, we developed a Java-based software, *TimeML2RDF*, that maps a temporal expression tagged with (a subset of) TimeML to a customizable set of RDF triples. In particular, the *target vocabulary* used by TimeML2RDF is completely configurable by defining *mapping rules*, declaratively stated in a separate file.

Before presenting the mapping rules, we briefly introduce HERO (Historical Event Representation Ontology), a computational ontology written in OWL2, developed within the Harlock'900 project. It includes the following modules:

- HERO-TOP is the top-layer of the ontology, linking it to the DOLCE foundational ontology (Masolo et al., 2003).
- HERO-EVENT provides a class hierarchy for events classification and properties for linking an event to its participants, to the geographical place where it occurred, and to the time interval in which it occurred.
- HERO-PLACE provides a characterization of geo-referenceable entities.
- HERO-TIME models time intervals, following Allen's algebra (Allen, 1983).
- HERO-ROCS defines the semantics of roles, organizations, collections and sets.

A complete description of HERO is out of the scope of this paper: we just sketch the main features of HERO-TIME, and quickly mention the classes and properties of HERO-EVENT used in the prototype. Figure 2 provides a (simplified) overview of the main classes and properties in HERO-TIME and their relations with elements in HERO-EVENT.

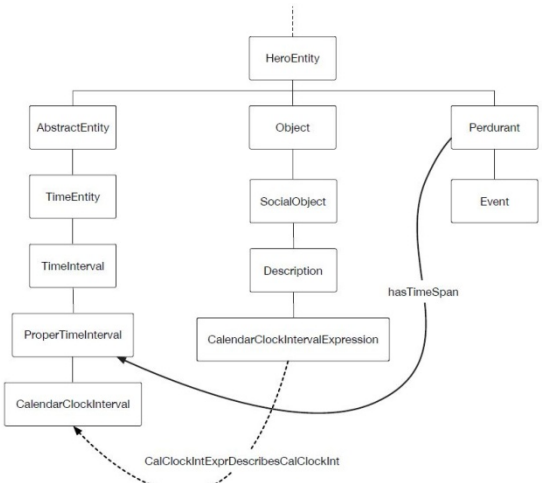

Figure 2: Events and time intervals in HERO.

Following Allen's interval algebra (Allen, 1983) HERO-TIME adopts a representation of time based on *time intervals* (or *time spans*), enabling reasoning about time at various levels of granularity and taking into account the inherently under-specification of temporal expressions typically found in historical sources (e.g., "at the beginning of 1945").

A time interval in HERO is characterized as an instance of the *TimeInterval* class – or, more often, of its sub-class *ProperTimeInterval*, representing time intervals with non-zero duration; *CalendarClockInterval*, sub-class of *ProperTimeInterval*, represents intervals referring to some calendar or time convention (e.g., years, months, days). Moreover, HERO-TIME models the distinction between a time interval (e.g., an instance of the *CalendarClockInterval* class) and the *expression* denoting it (e.g., an instance of the *CalendarClockIntervalExpression* class).

The main properties defined in HERO-TIME are based on the 13 basic relations described by Allen and representing all the possible relations between two time intervals. The most relevant properties in HERO-TIME are the following (in brackets the corresponding relation in Allen's algebra): *intStarts* (*starts*); *intEnds* (*finishes*); *intIn* (*starts* ∨ *finishes* ∨ *during* ∨ *equals*); *intBeginsIn* and *intEndsIn* (the relation between a time interval *t* and other two time intervals, t_b and t_e, that contain, respectively, the beginning and ending "instants" of *t*).

The HERO-EVENT module defines the class *Event* (sub-class of *Perdurant*, corresponding to the same class in DOLCE) to represent entities that happen in time. As depicted in Figure 2, the property *hasTimeSpan* associates a perdurant to a time interval, with the following meaning: given an event (perdurant) *e* occurring exactly in the time interval t_e, *hasTimeSpan(e,t)* implies that *intIn*(t_e,*t)*; in this way we account for the intrinsic "fuzziness" (under-specification) of time expressions used in natural languages (e.g., when we say that "Archduke Franz Ferdinand of Austria was killed on 28 June 1914" it is clear that we do not mean that the event lasted the whole day, but that it occurred in a shorter time interval contained in that day).

In order to translate TimeML tags produced by the Information Extraction module into RDF statements, different vocabularies (ontologies) can be used, by defining different mapping rules. Each rule is composed of two parts:

- a *matching section*, that defines a pattern of features of a TimeML tag;
- a *mapping section*, that contains all the information needed to produce the corresponding set of RDF triples.

For example, the following TimeML tag:

```
<TIMEX3 tid="t01" value="1943-09-10"
type="DATE">il 10</TIMEX3>
```

matches the rule labeled *YYYY-MM-DD date (day)*, that – thanks to its mapping section – produces the following RDF triples (ids have been simplified for the sake of readability):

```
<https://w3id.org/dataset_900/
   resource/time/id1>
a hero-time:Day ;
rdfs:label "1943-09-10" .
<https://w3id.org/dataset_900/
   resource/time/id2>
a hero-time:DoMMYDate ;
hero-time:accordingToCalendarType
   hero-time:gregorianCalendar ;
hero-time: calCIExprDescrCalCI
   <https://w3id.org/dataset_900/
     resource/time/id1> ;
hero-time:hasXMLSchemaGDateLex
   "1943-09-10"^^xsd:date .
```

The produced triples say that the identified time interval (*id1*) is a day (*hero-time:Day*) and the expression denoting it (*id2*) is a date consisting of three parts: day, month and year (*hero-time:DoMMYDate*), according to the Gregorian calendar (the *hero-time: calCIExprDescrCalCI* property associates the expression, *id2*, to the described time interval, *id1*; the *hasXMLSchemaGDateLex* property connects the

date *id2* to its lexicalization as an XML Schema *xsd:date*, i.e., "1943-09-10").

As far as the proof-of-concept prototype is concerned, the relevant events had been manually extracted, and the corresponding RDF triples manually generated, in a previous work (Caserio et al., 2017): therefore, we simply linked (through the HERO *hasTimeSpan* property) the already available semantic representation of such events to the RDF triples describing time intervals produced by TimeML2RDF. In this way, the final triplestore contains the information about the (relevant) events mentioned in the input texts, together with a fine-grained representation of the time interval in which they occurred, and the references to the documents in which the events are mentioned. This knowledge is available for third-party applications through a SPARQL endpoint.

3.4 Query Support Module

The temporal description of the events in the triplestore, while completely characterizing them, does not allow an efficient retrieval based on temporal range queries, due to different factors:

- Accessing the triplestore using SPARQL does not specifically support temporal queries;
- The time intervals contained in the triplestore have non-homogeneous granularities (spanning from hours to several years);
- We cannot practically use, at run-time, an OWL reasoner to infer additional temporal information from the triples, due to the well-known huge computational effort needed.

However, temporal relations defined in HERO enable us to order intervals that cannot be placed on a calendar reference system (e.g., "between December 1943 and February 1944") by stating their relation with calendar intervals.

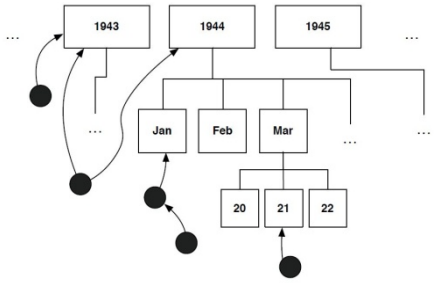

Figure 3: Data structure for temporal range queries.

Figure 3 depicts, with some abstraction, the indexing data structure that encodes the mentioned relations, i.e., essentially a tree representation of a calendar system where nodes represent either:

- Intervals with a 1:1 mapping to a calendar/ clock system (labeled boxes in Figure 3), corresponding to *CalendarClockInterval* instances, directly added to the data structure.
- Intervals related to a calendar interval via HERO relations (black circles in Figure 3), i.e., generic *ProperTimeInterval* instances, added to the data structure by linking them to a calendar interval through a HERO property.

Thanks to the described data structure, a temporal range query generates a tree visit between the two nodes delimiting the range, and the result will contain both calendar intervals and intervals specified using HERO relations.

Moreover, in many cases, it can be useful to include in the result not only intervals completely falling within the range, but also intervals partially overlapping with it. For example, given the range (selected by the user) spanning from 1942-12-04 to 1944-03-21, the time interval December 1942 has a partial overlap with the query range, so an event occurred in December 1942 ("fuzzy" specification) *may* have occurred during the time-span selected by the user. Events occurred in such time intervals can be useful for a user querying the knowledge base and can be retrieved by the Query Support module, thanks to the indexing data structure just described. This is particularly relevant in the historical domain because, frequently, events described in historical sources are placed in time in a rather "fuzzy" way and at different granularities, depending on their relative importance, the required precision, the author's preference, etc.

The knowledge encoded in such a data structure is made available via a RESTful web service that returns a serialized tree of the relevant time intervals IRIs (linked to events, in turn linked to historical documents) encoded in JSON.

4 EXAMPLE APPLICATION

As an example of application exploiting the rich semantic characterization of historical events and their temporal information, we developed a web-based UI enabling the visualization and exploration of historical events using temporal range queries. The application exploits the RESTful web service provided by the Query Support module to access the knowledge about historical events and their temporal properties and its ultimate goal is to provide users

with a smart access to the historical documents where the events presented in the UI are narrated.

The UI enables the user to query the application for events occurred in a given time period specified by start and end time intervals. To support both fine and coarse-grained searches, we allow the user to express the query with different granularities. Figure 4 shows the input control and the interaction sequence required to specify one extreme of the query range. In order to keep the UI as intuitive as possible, the control allows to specify only start and end dates having the same temporal granularity.

Figure 4: UI for the specification of the query range.

The control starts in an empty state (1), prompting the user to input a year; when the user has inserted a valid year (2) the control shows a "+" button that allows her to add the month granularity; she can proceed to specify a month (3) using the date-picker appearing below the control. The same steps can be repeated (5, 6) if the user wants to specify the granularity of a day.

Historical time is often represented in graphical form using timelines, i.e., geometrical mappings from time to space. Timelines can offer an effective overview of events in time and, when employed with uniform timescales, they can provide an insight on the relative duration of different events. However, we decided to discard a timeline-based visualization in our UI, mainly because the temporal data in our prototype have very different granularities and the described events are usually not related to a precise time duration; for example, an event linked to December 1943 and one linked to 1943-12-21 do not have comparable relative durations. For this reason, a simple geometrical mapping can be misleading, since it can transmit the wrong mental model to the user. Therefore, we opted for a representation based on a chronological list of events, where events do not possess any geometric dimension, but they are simply chronologically ordered. This ordering plays well with the vertical scrolling direction of a web application UI compared to the horizontal orientation typical of timelines. The resulting UI is shown in Figure 5, that displays a portion of the results obtained querying the system for events occurred in the period 1943-1945.

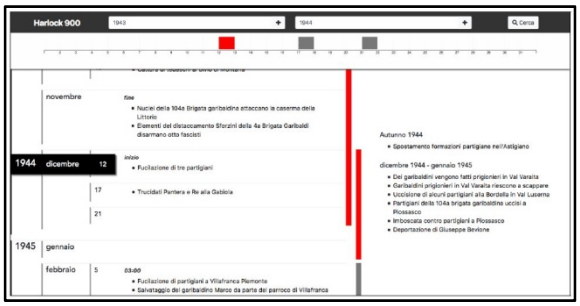

Figure 5: Result page for the query 1943-1945.

On the top of the page there is a menu showing the query made by the user. From these controls the user can edit the temporal range and start a new search. Just below there is a dynamic timeline that allows the user to have a better overall grasp of the time interval absolute position in the calendar and of the relative distance with other intervals.

In the central part of the page there is the chronological list of events occurred in time intervals that can be positioned on a calendar. Besides being chronologically ordered from top to bottom, the events are horizontally aligned with a calendar date (in year, month, or day granularity) rendered on the left and allowing the anchoring of the events in time. The hierarchical relation between time granularities of the calendar is rendered using spatial nesting from the left to the right.

The user can navigate through the list of events by vertically scrolling the page: the calendar and the list of events move outside the viewport to reveal new portions of the results. When a calendar element crosses the middle of the viewport, it is highlighted.

On the right-hand side of the central portion of the page there is a graphical representation of time intervals not expressed as *CalendarClockInterval* instances and defined as time spans between two calendar intervals (characterized using the *intBeginsIn* and *intEndsIn* HERO properties). Each interval is displayed as a rectangular bar, in which the top and the bottom sides are horizontally aligned with the start and end calendar dates delimiting the interval. This section scrolls together with the calendar and the events list. When an interval bar intersects the middle of the viewport, the events occurred in that time interval are listed on the right. When the user pointer hovers over an event description, the bar denoting the corresponding time interval is highlighted, isolating it from other possible overlapping bars.

Finally, the application exploits the possibility that some events are characterized as occurring in a time interval that only partially overlaps with the query interval. In these cases we cannot be sure whether or not to include these events in the result set, so they are displayed with an appropriate graphical distinction (italic font-face).

The application presented in this section demonstrates:

- The possibility, for external applications, of exploiting the knowledge provided by our pipeline (namely, the RESTful web service offered by the Query Support module).
- The advantages of having a rich semantic representation of the content of historical resources (namely, in the current prototype, events with a fine-grained representation of temporal information).

5 CONCLUSIONS

The proof-of-concept prototype presented in this paper shows the feasibility and the effectiveness of our approach. However, in order to fully assess it, we plan to integrate both the back-end and the UI within a wider application, enabling users to navigate – by querying and browsing – historical events narrated within archival documents. Such an application, besides time information, will take into account places where events occurred (i.e., geographical information) and event participants, together with their roles in the events. The back-end integration will be supported by the HERO ontology and the usage of Semantic Web standards, while the integration of the UI represents an interesting challenge, which is under study.

ACKNOWLEDGEMENTS

This work has been partially supported by Università di Torino and Compagnia di San Paolo within the PRiSMHA project. We are grateful to M. Rovera for the valuable suggestions on IE from historical texts.

REFERENCES

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., 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.

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

Boschetti, F., Cimino, A., Dell'Orletta, F., Lebani, G. E., Passaro, L., Picchi, P., Venturi, G., Montemagni, S., Lenci, A. 2014. Computational Analysis of Historical Documents: An Application to Italian War Bulletins in World War I and II. In *LREC 2014 Workshop*.

Caselli, T., Sprugnoli, R., 2015. *It-TimeML - TimeML Annotation Guidelines for Italian, v. 1.4*. Technical Report. VU Amsterdam and FBK.

Caselli, T., Sprugnoli, R., Speranza, M., Monachini, M., 2014. Eventi evaluation of events and temporal information at EVALITA 2014. *4th EVALITA*, 27-34.

Caserio, M., Goy, A., Magro, D., 2017. Smart access to historical archives based on rich semantic metadata. In *IC3K – KMIS'17*. INSTICC SciTePress, 93-100.

Cybulska, A., Vossen, P., 2011. Historical Event Extraction from Text. In *LaTeCH'11*, 39-43.

Ehrmann, M., Colavizza, G., Rochat, Y., Kaplan, F., 2016. Diachronic evaluation of NER systems on old newspapers. In *KONVENS'16*, 97-107.

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

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., D'Ambrosio, M., 2017. PRiSMHA (Providing Rich Semantic Metadata for Historical Archives). In *CREOL 2017*.

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

Heath, T., Bizer, C., 2011. *Linked Data: Evolving the Web into a Global Data Space*, Morgan & Claypool.

Hogenboom F., Frasincar F., Kaymak U., de Jong F., 2011. An Overview of Event Extraction from Text. In *DeRiVE'11 at ISWC 2011, Vol. 779*.

Manfredi, G., Strötgen, J., Zell, J., Gertz, M., 2014. HeidelTime at EVENTI: Tuning Italian resources and addressing TimeML's empty tags. In *4th EVALITA*, 39-43.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., 2003. *WonderWeb Deliverable D18*. Technical Report, CNR.

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

Moretti, G., Sprugnoli, R., Menini, S., Tonelli, S., 2016. ALCIDE: Extracting and visualising content from

large document collections to support humanities studies. *Knowledge-Based Systems*, 111, 100-112.
Mulholland P., Wolff A., Kilfeather E., 2015. Storyscope: Supporting the authoring and reading of museum stories using online data sources. In *WebSci 2015*, ACM Press.
Pustejovsky, J., 2017. ISO-TimeML and the Annotation of Temporal Information. In Ide, N., Pustejovsky, J. (Eds.), *Handbook of Linguistic Annotation*, Springer, 941-968.
Rovera, M., Nanni, F., Ponzetto, S. P., Goy, A., 2017. Domain-specific Named Entity Disambiguation in Historical Memoirs, In *CLiC-it'17, vol. 2006*. CEUR.
Segers, R., van Erp, M., van der Meij, L. 2011. Hacking History via Event Extraction. In *K-CAP'11*, 161-162.
Sprugnoli, R., Tonelli, S. 2016. One, no one and one hundred thusand events: Defining and processing events in an inter-disciplinary perspective. *Natural Language Engineering*, 23(4), 485-506.
Strötgen, J., Gertz, M., 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2), 269-298.
Strötgen, J., Armiti, A., Van Canh, T., Zell, J., Gertz, M., 2014. Time for more languages: Temporal tagging of Arabic, Italian, Spanish, and Vietnamese. *ACM Transactions on Asian Language Information Processing*, 13(1), 1-21.