Baltic J. Modern Computing, Vol. 6(2018), No. 4, 449-463
https://doi.org/10.22364/bjmc.2018.6.4.09

Semantic Annotation Tool for Cultural Heritage Content

Uldis BOJĀRS[1,2], Anita RAŠMANE[1], Artūrs ŽOGLA[1],
Signe BĀLIŅA[3], Edgars SALNA[3]

[1] National Library of Latvia, Mūkusalas iela 3, Rīga, LV-1423, Latvia
[2] Faculty of Computing, University of Latvia, Raiņa bulvāris 19, Rīga, LV-1459, Latvia
[3] SIA "Datorzinību centrs", Lāčplēša iela 41, Rīga, LV-1011, Latvia

{uldis.bojars, anita.rasmane, arturs.zogla}@lnb.lv;
{signe.balina, edgars.salna}@dzc.lv

Abstract. This paper describes the architecture and implementation of a semantic text annotation tool for cultural heritage content. The requirements for this tool are based on text annotation case studies at the National Library of Latvia and were generalized to be applicable to a wider range of annotation projects. The tool implements a rich and flexible annotation model with support for three core types of annotations (simple, composite and structural), user-definable annotation and entity classes, and advanced functionality such as links between annotations. Information about named entities referenced from annotations is collected in an integrated entity database, accessible using a Linked Data interface.

Keywords: Semantic annotation, Linked Data, Cultural heritage, Text annotation, Text enrichment

1. Introduction

Cultural heritage organizations – libraries, archives, museums – are in charge of large amounts of information, including collections of textual content ranging from scans and transcriptions of ancient documents to modern, digitally-born documents.

Digital documents make it possible to add annotations to these documents and to apply natural language processing and data mining methods that help users discover new information patterns and extract new knowledge from these documents. One of the ways for analyzing and enriching text content is to identify the facts, objects and other useful information mentioned in these documents. Researchers and general public may be interested in recording different kinds of information contained in text documents and in marking up important areas of text, identifying mentions of named entities and other knowledge included in or related to the document.

There is a substantial amount of previous research done in Named Entity Recognition (NER) aimed at identifying and marking up mentions of objects of selected types in natural language text documents (Atdağ and Labatut, 2013). Most of the current tools can be trained to recognize the basic types of objects: Persons, Organizations, Places, Dates, etc. However, users may need to mark up additional types of information that is specific to a particular collection of text documents. A biology researcher, for example,

may want to annotate mentions of Plants, Insects and Mammals in their research documents. This task would require NER tools that can be recognize user-defined object types and information about the entities from the domain of interest.

Cultural heritage content and especially historical documents may be particularly difficult for automatic entity recognition and linking because the relevant tools need to know the specific context of the documents (e.g. personal correspondence and people involved in it) and entities that are likely to be mentioned in these documents. Another issue is the need to disambiguate between different meanings of the same text fragment which requires understanding the document and its context, and may require in-depth investigation by domain experts (c.f. an example described in Section 4). In these cases, when there are difficulties getting high-quality named entity annotations automatically, users have to resort to manual or semi-automated annotation.

In the initial stage of this research, we analyzed text annotation needs of cultural heritage document collections at the National Library of Latvia (NLL) and identified the requirements and the annotation model for this type of information (Bojārs et al., 2017). Our work focused on manual text annotation and the annotation scenarios that are involved in annotating and enriching cultural heritage content such as the information included the linked digital collection "Rainis un Aspazija" (Bojārs, 2016). In particular, we proposed a flexible model involving multiple types of annotations, user-customizable annotation and entity classes, and an integrated Entity database for recording information about the objects that annotations refer to.

This paper describes the annotation tool developed based on the annotation model and requirements developed earlier in the project. It is a functional prototype that supports various scenarios for manual text annotation by domain experts.

The paper is organized as follows: Section 2 contains a summary of the annotation model and requirements; Section 3 describes the architecture and implementation of the prototype; Section 4 provides a demonstration of the tool in action; Section 5 discusses related work; and Section 6 concludes the paper.

2. Annotation requirements

Text annotation tasks supported by the annotation tool include various annotation scenarios ranging from simple highlighting to more complex use cases:

- Highlighting a text fragment (adding visual display information to text);
- Adding comments to text fragments (e.g. for saving notes to be used in subsequent annotation stages);
- Assigning annotation classes to text fragments (using annotation classes for distinguishing between mentions of different types of objects such as Persons or Locations);
- Identifying the entities mentioned in text fragments by linking text fragments to the unique identifiers for these entities;
- Describing more complex information that may be represented by multiple text fragments (e.g. an Event along with text fragments describing its details).

All these different scenarios may come up in typical annotation tasks. For example, a researcher may be annotating personal correspondence by first highlighting text

fragments of interest (i.e. creating simple annotations for these text fragments) and adding initial comments to them. If the researcher already knows what type of entity this text fragment is about, they would add this information as well.

Identifying the exact entities or concepts referred to by these text fragments is a complex task requiring further exploration and the researcher may decide to return to this task later on. When ready, they would identify the entity mentioned in the text fragment and enrich the annotation with a unique identifier for this entity. Often the researcher will have collected additional information about this entity (including information that sets it apart from other similar entities). The annotation tool needs to make it possible to preserve this information.

Users may also need to record more complex information such as events mentioned in the text. The individual text fragments, describing different aspects of these events – date, location, participants, may have been already annotated in the previous steps. However, an "umbrella" annotation is still needed to define that this is an *Event* and to link these annotations together. The tool needs be flexible and allow users to define such annotations, including the ability to define new annotation properties (e.g. for linking a *Theatre Performance* event to the *Work* being performed).

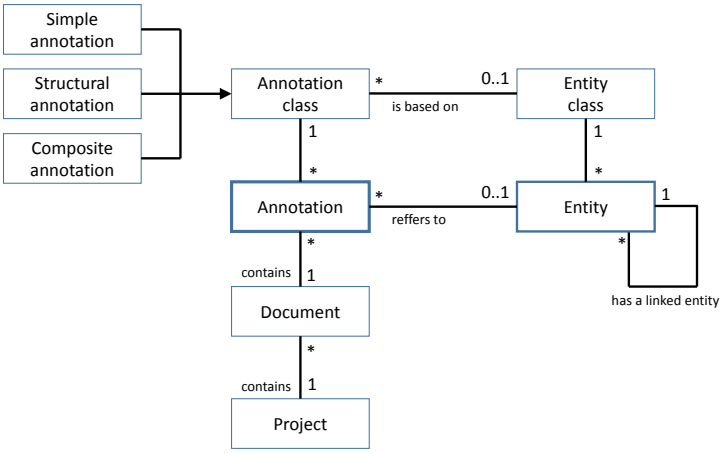

Figure 1. Conceptual model of the document annotation tool.

An overview of the annotation model implemented in the annotation tool prototype is shown in Figure 1:

- *Documents* contain the textual content to be annotated and are organized in *Projects*;
- *Annotations* attach some information to parts of the *Document* (text fragments);
- *Text fragments* are parts of documents (e.g. one or more words) that annotations are attached to;
- *Annotation class* is a denomination specifying the type of the *Annotation* that the text fragment refers to (e.g. Poet, River, Jewelry). Every annotation belongs to a single *Annotation class*.
- *Annotations* belong to one of three *core annotation types*: Simple annotations, Structural annotations or Composite annotations.

- *Entities* are distinct, identifiable objects mentioned in the text. An *Annotation* may include a reference to the relevant *Entity* (in case if this entity has been identified and recorded in the annotation tool).
- Information about *Entities* is recorded in the *Entity database.*
- *Entity classes* are used to distinguish between different types of entities (e.g. Person or Location). Every entity belongs to a single *Entity class*.

The *Entity database* contains all relevant information about entities referenced from annotations. While annotation *Projects* may be private (and have access right management associated with them), the entity information is shared across projects. Entities may have qualified links relating them to other entities in the database (e.g. Spouse) as well as references to additional, external Linked Data or web resources about this entity.

The system supports user-defined annotation classes and entity classes. Every *Annotation class* is based on a related *Entity class* (e.g. annotations of class *Poet* may be based on and refer to the entity class *Person).*

Annotations may have *Annotation properties* that are key / value pairs where value is either a text string or a reference to another annotation (e.g. Location). *Annotations* also have comments and technical metadata related to the annotation process (e.g., creator and creation time of annotation).

Most annotation scenarios (e.g. adding comments or referencing entities mentioned in the text) are covered by Simple annotations. These annotations may contain a reference to an entity in the Entity database. Structural annotations are used for marking up parts of the text that do not refer to any entity but have a special meaning in the context of the document (e.g. an interjection in parliamentary session transcripts).

Composite annotations are used for more complex use cases such as annotating Events. These annotations typically reference other annotations associated with text fragments that describe details of the composite annotation.

Requirements for semantic annotation were identified during two research project activities – (1) analysis of current annotation trends and annotation tools, and (2) case studies of practical cultural heritage annotation needs at NLL. The case studies were based on two datasets: correspondence (letters) from the late 19th century between two famous Latvian poets (Rainis and Aspazija) and Parliamentary transcripts that document the first four parliamentary terms in Latvian history (1922-1934). The case studies were performed by NLL domain experts in collaboration with developers of the annotation tool.

During these case studies we concluded that the available annotation tools do not fully satisfy NLL's annotation needs, identified annotation requirements, summarized in this section, and developed the annotation tool prototype that implements these requirements. We identified over 30 high-level annotation requirements, divided into following major areas: annotations, entities, annotation process, interoperability and user interface.

Annotation requirements identify a need for annotation classes, which should be standardized (place, person, organization, date, time etc.) but users should also be able to implement and configure additional classes in order to make the tool suitable for various domains. Annotations may overlap and a single document may contain annotations created by multiple users. As described above, annotations may reference other annotations. Users should also be able to add comments to the annotation. The tool

should include support for suggesting new annotations based on entity information and on the information about previously created annotations.

The next group of requirements is related to entities and the Entity database. The annotation tool should allow users to annotate mentions of entities in the text. It should be possible to link entities with authoritative dictionaries, as well as to export entity information as Linked Data. The Entity database should allow users to find for entities and manage their information.

An important requirement for the annotation process is the persistence of annotations with respect to changes in the text that is annotated (e.g. the original text may change due to the necessity to correct errors in scanned and OCR-recognized text). In order to be sufficiently robust to such changes, the tool should be able to update annotation data or, if the loss of existing annotation positions cannot be avoided, to notify users about unrecoverable changes.

Another important requirement group was related to automation of annotation process by detecting the potential text fragments to be annotated. This automation may be achieved by using the already existing annotations from documents similar to the one being annotated. Similar documents can be grouped together in Projects.

Also, significant attention was given to interoperability aspects of the tool. Interoperability requirements include annotation import and export, the use of URI identifiers for annotations and entities, and providing the ability to publish annotated document sets or projects on the Web.

Additional information about the annotation model and annotation requirements can be found in Bojārs et al. (2017).

3. Architecture for Semantic Annotation

The architecture of the semantic annotation tool was developed according to the requirements identified in the initial phase of the research project. The technical architecture is based on a three-layer architecture consisting of the application, services and a database. The application layer (user interface) is implemented as a web application using Angular framework (version 4.3.1) in order to achieve a responsive and flexible user interface.

In order to enable the user interface to be as independent as possible and to facilitate the implementation of alternative user interfaces, all interaction with data objects is provided by REST web services that use JSON with an application-specific data model. Data storage is implemented using MS SQL server relational database.

While the main requirements for the annotation tool were a part of earlier research described in Bojārs et al. (2017), the prototype development stage helped us identify additional requirements not only for the annotation process but also for document, project and user management.

3.1. System support for the annotation process

Document annotations are organized in projects – sets of contextually related documents. These documents contain similar textual content and may use the same annotation classes and properties.

Every annotation belongs to a single annotation class which defines the properties that annotations of this class may have. While annotation classes are specific to a Project, their properties are similar to entity class properties. Therefore, the annotation tool supports transferring entity class properties to a related annotation class. This provides guidance for every project as their entity classifications will be passively guided by the same entity classification structure.

Despite similarities and properties inheritance, annotation class has some individual properties, such as the core annotation type and user-defined properties. The annotation tool has three core annotation types (specialisations): "Simple annotations", "Structural annotations" and "Composite annotations". Based on specialisation property value, the system changes the functional behaviour of annotations marked with a specific annotation class. The user-defined properties of annotation class serve as a template for entering annotation information. As a result, the annotations with the same annotation class tend to have similar properties. This annotation class implementation allows us to reduce manual input during the annotation process yet allows for high flexibility for classification used in different projects.

When the annotation project and its initial annotation classification have been registered, the system allows users to add documents and to start annotating them by choosing a document and marking up text fragments to be annotated. In order to support project-based, multi-user annotation the system contains additional functionality such as document status control and document access control.

The prototype also provides initial support for advanced annotation functionality: preserving document annotations when the document text changes (if updated document versions are added) and automatic generation of annotations based on earlier annotations registered in other documents in the project.

3.2. Document annotation user interface

In order to make the time-consuming annotation process easier and to meet the usability expectations of users, special attention was devoted to developing a user interface that supports a fluent annotation process. User interface requirements were defined by NLL researchers and software engineers of Datorzinību centrs. The goal of this joint work was to create an annotation tool with a user interface which is suitable for key annotation use cases and implements the requirements identified during research of existing annotation tools and accumulated experience from NLL dataset annotation case studies. In addition to these research activities, the agile development process involved user experience test sessions (practical annotation work) aimed at identifying the necessary user interface improvements.

A schematic representation of the user interface is shown in Figure 2. The reasoning for chosen element positioning is based on the following considerations:

- users tend to work on annotation task using large screens or even two screens;
- document text is considered of the most importance, as a result a half of the interface space is dedicated to the document text;
- there must be a visible list of annotations that have been created in the document;
- there must be advanced highlighting and interaction functionality that connects annotated document text fragments and the list of annotations;

- there must be easy to use annotation filtering solution with possibilities for customisation;
- there must be a fast and easy way to view and edit information about an individual annotation (annotation class, entity reference, annotation properties).

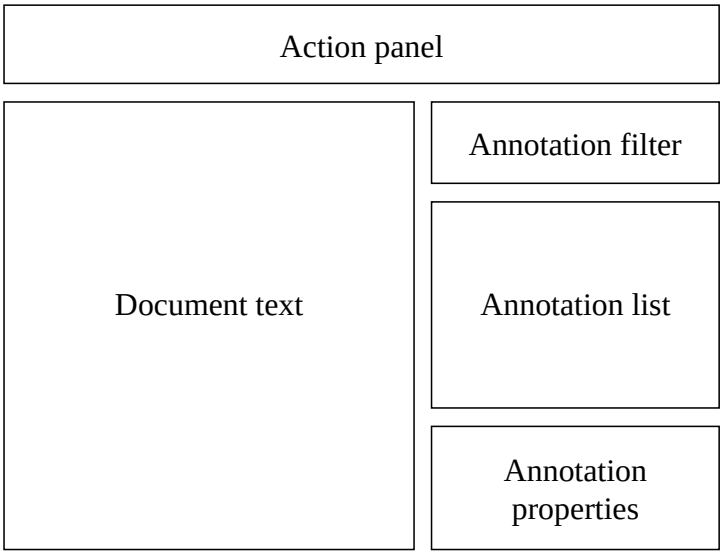

Figure 2. Layout of the document annotation user interface.

In order to meet the requirement for advanced highlighting and interaction, all of the annotations that are present in the annotation list (according to the current annotation filter) are highlighted in the document text using colour patterns defined by annotation classes. Additional interaction between highlighted text fragments and annotations in the annotation list is implemented as a two-way position synchronisation. If the user selects highlighted text in the document, the system will select the corresponding annotation in the annotation list and vice versa. Additionally, information about the selected annotation will be shown in the annotation properties view below the annotation list.

This interface structure allows easy navigation of document annotations and fast access to chosen annotation properties. In order to support composite annotations and the ability to create references to other annotations in the same document, the prototype includes functionality for adding links between annotations.

Users add new annotations by entering information in the annotation properties view, entering at least annotation core type and class. In order to enter information about the entity represented by the text fragment, the properties view offers functionality for looking up existing entities in the Entity database and examining their details. In case if the required entity is not in the database, users are able to add a new entity and continue with the annotation process. Additional information that can be entered includes annotation properties (incl. links to other annotations) and comments.

3.3. Annotation tool prototype

The architecture for semantic annotation, described in this paper, was implemented as a prototype annotation tool. It is deployed at the National Library of Latvia (NLL) and is accessible to researchers that use library's services. Its architecture allows integration with third party software via an API and adding specific functionality and customizations necessary for the end-user. The tool has multi-language support (currently available UI languages are English and Latvia). While the prototype was built to support semantic annotation needs of cultural heritage organizations, it is also suitable for other application areas.

A screenshot demonstrating the tool is included in Section 4. The development of the prototype was funded by of the program for promoting long-term cooperation between ICT enterprises and science institutions, and creating prototypes of competitive IT products intended for later commercialization. The resulting tool can be obtained from its developer (Datorzinību centrs[1]) as a standalone installation or using the Software as a Service model.

3.4. Interfaces and formats for annotation data exchange

The prototype can export and import annotation information using an application-specific JSON data format that represents information about documents, their annotations and entities referenced from these documents.

Initially, we envisioned data exchange to use an adapted version of the W3C Web Annotation format as described in Bojārs et al. (2017). However, this format does not cover all the internal information that is necessary for saving annotations. As a result, the tool has its own JSON-based format that allows us to fully represent information about the annotation state. Users can also use the system's publishing functionality to export a Web view of collections of annotated documents that are standalone, read-only versions of the document annotation user interface.

A limitation of the current prototype is that it focuses on the core annotation process and does not include functionality for analysis of resulting annotation data. This functionality can be implemented separately using exported annotation information. In order to ensure interoperability, we plan to add additional data export formats such as TEI (TEI Consortium, 2018).

```
<https://anotators.lndb.lv/entities/1038>
    a <https://anotators.lndb.lv/entitytypes/2> ;
    dc:type "Vieta" ;
    foaf:name "Dunavas pagasts"@lv ;
    foaf:name "Dunava parish"@en ;
    a geo:SpatialThing ;
    geo:lat "56.1962" ;
    geo:long "26.179" ;
    foaf:page <https://lv.wikipedia.org/wiki/Dunavas_pagasts> ;
    foaf:page <http://www.zudusilatvija.lv/objects/object/15254> ;
    owl:sameAs <https://www.wikidata.org/wiki/Q3801462> ;
    owl:sameAs <http://www.geonames.org/11352634> .
```

Listing 1. Linked Data representing an Entity database entry.

[1] http://www.dzc.lv/en/

Prototype's Entity database contains all entity-related information including entity class, labels (including their language tags), comments, relations to other entities and external links to both web and Linked Data resources. Entity database is global across annotation projects. All entities have URI identifiers and their information is published as Linked Data (Berners-Lee, 2009). Listing 1 shows an abbreviated version of information about the Dunava parish, an entity of class Location. The information shown in the listing includes entity types, its name in two languages, geographical location (in the WGS84 coordinate system[2]), and links to external Linked Data resources (on GeoNames and Wikidata (Erxleben et al., 2014)) and web pages about this entity[3]. Users can retrieve additional information about these entities by following Linked Data links to external resources such as Wikidata[4].

Linked Data interface could have also been developed for the information about annotations. However, since prototype's projects and documents are not public (though they can be exported and published on the web) and are protected by user access rights, it did not make sense to include them in the Linked Data interface at this stage.

4. Annotation Prototype in Use

This section demonstrates the developed prototype and highlights some of the challenges associated with the annotation process. The document used in this demonstration comes from the use case of the linked digital collection "Rainis and Aspazija" (Bojārs, 2016) and is an annotated version of the abstract of Aspazija's play "Aspazija".

A screenshot of this document in the annotation tool is shown in Figure 3. The left side of the workplace contains the annotated document with the list of document annotations appearing on the right-hand side. These annotations belong to the user-definable classes of annotations such as Concept, Person, Play and Theatre. Each class of annotations is shown in the document in a different colour.

We selected this example to show the difficulties of semantic differentiation of named entities to be found in text. The short text of the document contains annotations from annotation classes *Person* and *Character* (entity class *Person*), *State* (entity class *Location*), *Publisher* and *Theatre* (entity class *Institution*), *Novel* and *Play* (entity class *Work*), *Occupation* (profession) and *Concept* (entity class *Term*), and *Date* (entity class *Time*).

Even the short title of this usage case highlights the problem of correct semantic identification of entities represented by identical or very similar text fragments, whose essentially different nature is rather easily seen by a human researcher but can be very hard for automated recognition, marking and annotation. Most of the entities mentioned in text are well known and easy identifiable, apart from a notable exception of "Aspazija". This text contains 5 different entities that can be represented by text fragment *Aspazija*:

---

[2] https://www.w3.org/2003/01/geo/

[3] The URI for this entity refers to an instance of the annotation tool prototype deployed at the National Library of Latvia. Access to this instance is limited due to operational reasons, therefore Linked Data about its resources is only available on NLL's network.

[4] https://www.wikidata.org/wiki/Q3801462

- Latvian poetess Aspazija (1865-1943);
- Classical-era Athens Aspasia (470-410 BC), written in Latvian as Aspazija;
- Aspazija's play "Aspazija";
- novel "Aspazija" by Austrian writer Robert Hamerling (1830-1889); and
- the character Aspazija in the play "Aspazija".

Figure 3. Screenshot of the annotation editing workplace with document text and annotations.

The annotation tool allows users to mark all mentions of "Aspazija", to assign meaningful annotation classes to every one of them (two *Persons*, one *Character*, one *Play* and one *Novel*) and to create and maintain references to corresponding entries in the Entity database. The document annotation interface allows users to create and edit annotation classes and entries in the Entity database "on the fly". The information in the Entity database is also available for editing independently. This means that a researcher could create entries for his or her project before starting the annotation process.

When linking annotations to entities, researchers have the option to choose which of the entity's labels (which can be in multiple languages) to display in this annotation. An example of an entity name in a language different from the main text, German, can be seen in the list of annotations on Figure 3: "Aspasia : Ein Künstler und Liebesroman". This example illustrates researcher's choice to emphasize the original title by choosing entity name in German among other names in the Entity database.

The next step in the annotation process facilitated by the annotation tool is defining Composite annotations – semantic "sentences" composed of groups of related annotations. In the case of five annotations related to different text fragments "Aspazija" we can define the following relations among annotations, expressed here in a free text form:

[*Person*] Aspazija (1865-1943) [*Property*] "is an author of" [*Play*] Aspazija;
[*Person*] Robert Hamerling [*Property*] "is an author of" [*Novel*] Aspazija;
[*Play*] Aspazija [*Property*] "is written by" [*Person*] Aspazija (1865-1943);
[*Novel*] Aspazija [*Property*] "is written by" [*Person*] Robert Hamerling;
[*Person*] Aspazija (470-410 BC) [*Property*] "is prototype for" [*Play*] Aspazija;
[*Person*] Aspazija (470-410 BC) [*Property*] "is prototype for" [*Novel*] Aspazija;
[*Character*] Aspazija [*Property*] "is character in" [*Play*] Aspazija.

All this information is derived by human expert from the text of the document, except for life years for two persons with the same name, which are used for disambiguation purposes. The relations linking different annotation classes are defined locally in the project. As a future development, they could be formalized, using either a simple local vocabulary or already developed metadata schemes and ontologies such RDA (Resource Description & Access), whose elements and properties are expressed by URIs[5]. RDA is a modern standard for representing bibliographic information in libraries and other cultural organizations and is one of the options for representing cultural heritage annotations.

The third type of annotations – Structural annotation – is used for marking large parts of document with different semantic importance, for example, "citation", "translation of previous part of text in a different language" or "one session of meeting". Regular (simple) annotations could be marked in any place of document, but inside a particular Structural annotation they may obtain a more specific meaning. Figure 3 contains a structural annotation "Citation" represented by five lines of text highlighted in light grey.

The outcomes of the full annotation process of this document are:

- Document annotated with different types of annotations (simple, composite, structural);
- New and enriched Entity database entries with entity names in different languages and links to authority records (e.g. VIAF), encyclopaedic on-line resources (e.g. Wikipedia), Linked Data resources (e.g. Wikidata and GeoNames), and digital libraries (e.g. LNDB Grāmatas);

5. Related Work

There are several tools available that perform Named Entity Recognition (NER). In some cases these tools perform only the classification of concepts. Stanford NER[6] has several NER models that can recognize from 3 up to 7 classes of objects. The tool works best on English texts, however, it also has support for German, Chinese and Arabic texts.

Figure 4 illustrates one drawback of a purely automatic recognition process. The second mention of the word "Flavian" has been incorrectly classified as a Location, although in this case it refers to a group of Persons. While such mistakes can be tolerable in some cases (depending on the purpose of the annotation project), it is not an option in our cultural heritage use cases where annotations in the digital collection have to be as precise as possible for this collection to be useful to other researchers.

[5] "RDA Registry", https://www.rdaregistry.info/
[6] https://nlp.stanford.edu/ner/

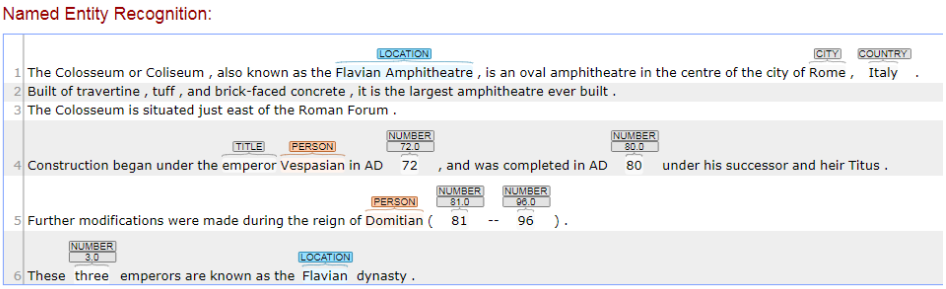

Figure 4. Results of automatic named entity recognition process (example).

Figure 4 illustrates one drawback of a purely automatic recognition process. The second mention of the word "Flavian" has been incorrectly classified as a Location, although in this case it refers to a group of Persons. While such mistakes can be tolerable in some cases (depending on the purpose of the annotation project), it is not an option in our cultural heritage use cases where annotations in the digital collection have to be as precise as possible for this collection to be useful to other researchers.

Another publicly available NER tool is the Dandelion API[7]. Besides classification of objects, Dandelion also identifies the entities mentioned in the text. Dandelion API supports NER for 6 classes of objects: Persons, Works, Organizations, Places, Events and Concepts. The API identifies objects by linking them to appropriate Wikipedia or DBpedia pages. It can be used if the text primarily contains mentions of entities that have Wikipedia pages but, like other automated NER tools not specifically aimed at historical content, it is not helpful for old texts where the entities are mentioned in specific context and may not be notable outside this context.

Figure 5. Example of NER of Latvian text with Dandelion API.

Dandelion API has Beta support for Latvian language. Figure 5 shows its result of named entity recognition on the abstract of Aspazija's work "Aspazija" mentioned

[7] https://dandelion.eu/

earlier. In this case it identifies entities quite well (however, this is modern text mentioning notable entities present in Wikipedia) although it fails in some trickier situations such as the several mentions of the word "Aspazija", which in all cases has been identified as the Latvian poetess Aspazija (1865-1943). In reality there are four different concepts mentioned here: Latvian poet Aspazija (1863-1943), Ancient Greek poet Aspasia (around 470-410 B.C.) and two different works titled "Aspazija". Identification of the correct entity in each case requires manual work by a domain expert.

The approach that we have chosen in order for the annotation tool to support the annotation of cultural heritage content (including highly contextual historical texts) is to help users perform manual text annotation which is a different task from automated annotation and NER approaches.

A popular text annotation that follows the manual approach is Hypothes.is – a web-based, collaborative open source tool for annotating web pages and PDF files available on the Web (Perkel, 2015). It has an intuitive annotation interface that allows users to highlight text, annotate it (by adding comments and tags) and to have discussions by replying to other annotations. Similar to our prototype, its user interface consists of the document visible on the left side of the screen and a sidebar with annotations on the right-hand side. Annotation information is kept on Hypothes.is servers and is available via an API. While it is a valuable text annotation tool, it does not support named entity annotations and other important requirements that are needed for our use cases.

The annotation tool prototype described in this paper supports the functionality required for our cultural heritage use cases that was not supported by the existing annotation tools we examined (e.g. user-defined annotation classes and entity classes; structural and composite annotations; an integrated entity database with a Linked Data interface).

6. Conclusion

This paper described the semantic annotation tool prototype developed for based on the annotation model and requirements for cultural heritage annotation use cases. This tool was developed for the annotation use cases of the National Library of Latvia. Due to the specifics of annotating context-dependent historical texts, the goal of the annotation tool was to support manual text annotation and make it as fluent for users as possible.

The annotation tool implements a rich annotation model (Bojārs et al., 2017) that supports three core types of annotations (simple, structural, composite), user-defined annotation and entity classes, annotation properties and links between annotations.

This tool allows users to annotate documents, organized in projects, and to maintain information about the entities mentioned in annotations. Its entity database contains all relevant information about these entities, including links to external sources describing these entities (such as Linked Open Data resources). Information about entities is available as Linked Data.

The prototype is deployed at the National Library of Latvia and is being used for enriching cultural heritage collecting including annotations for the next stage of the linked digital collection "Rainis and Aspazija".

Acknowledgements

The research leading to these results was funded by the European Regional Development Fund project "Competence Centre of Information and Communication Technologies", (1.2.1.1/16/A/007), individual research project 2.1 "Semantic annotation of textual data in web environment for related data sets". Work on this paper was partially supported by the University of Latvia project AAP2016/B032 "Innovative information technologies".

References

Atdağ, S., Labatut, V. (2013). *A comparison of named entity recognition tools applied to biographical texts.* In: 2nd International Conference on Systems and Computer Science (ICSCS), IEEE, 2013, pp. 228–233.

Berners-Lee, T. (2009). *Linked Data - Design Issues.* Revision: 18.06.2009. URL: http://www.w3.org/DesignIssues/LinkedData.html

Bojārs, U. (2016). *Case study: Towards a linked digital collection of Latvian Cultural Heritage.* Proceedings of the 1st Workshop on Humanities in the Semantic Web (WHiSe 2016). CEUR Workshop Proceedings, vol. 1608, pp. 21–26.

Bojārs, U., Rašmane, A., Žogla, A. (2017). *The requirements for semantic annotation of cultural heritage content.* Proceedings of the 2nd Workshop on Humanities in the Semantic Web (WHiSe 2017). CEUR Workshop Proceedings, vol. 2014, pp. 69–79.

Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D. (2014). *Introducing Wikidata to the linked data web.* In: International Semantic Web Conference, Springer, Cham, pp. 50–65, 2014.

Perkel, J. M. (2015). *Annotating the scholarly web.* Nature News, 528(7580), 153. URL: https://www.nature.com/news/annotating-the-scholarly-web-1.18900

TEI Consortium (2018). *TEI P5: Guidelines for Electronic Text Encoding and Interchange.* Version 3.4.0. Last updated: 23.07.2018. URL: http://www.tei-c.org/Guidelines/P5/

Authors' Information

Dr. Uldis Bojārs is an Associate Professor at the Faculty of Computing, University of Latvia and a Semantic Web Expert at the National Library of Latvia. His areas of interest include Linked Data, Open Data and Social Semantic Web. Uldis has a PhD in Computer Science from the National University of Ireland, Galway and is a co-founder of the SIOC Project aimed at applying Semantic Web technologies to Social Web sites.

Anita Rašmane is a Systems Librarian at the National Library of Latvia and the main editor of annotation information for the linked digital collection "Rainis and Aspazija". She has participated in Web harvesting and archiving projects at the National Library of Latvia, as well as several essential library data enhancement projects. Anita's main topics of interest are implementing new data models, improving data quality and the challenges of data user interfaces and publishing.

Artūrs Žogla is a Head of Digital Library at the National Library of Latvia. In this position he has been involved in development of several Digital Library systems and research studies on using Semantic Web technologies on library data. Artūrs Žogla has previously been a lecturer at the University of Latvia where he taught a course on Semantic Web basics.

Dr. Signe Bāliņa is a researcher at SIA "Datorzinību centrs" and Professor at the Faculty of Business, management and economics, University of Latvia. She has actively participated in information technology, finance, economics and education development fields. During the last 5 years she has contributed to many scientific publications, has co-authored three books and has contributed to seven scientific research projects.

Edgars Salna is a researcher at SIA "Datorzinību centrs" and a PhD student at the Faculty of Business, management and economics, University of Latvia. His background is in commercial software development and in research prototype development.

Received October 24, 2018, revised December 27, 2018, accepted December 28, 2018