The Requirements for Semantic Annotation of Cultural Heritage Content

Uldis Bojārs[1,2], Anita Rašmane[1] and Artūrs Žogla[1]

[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
[uldis.bojars,anita.rasmane,arturs.zogla]@lnb.lv

Abstract. This paper focuses on the semantic annotation of textual content and on annotation requirements that emerge from the needs of cultural heritage annotation projects. The requirements presented here are based on two text annotation case studies at the National Library of Latvia and were generalised to be applicable to a wider range of annotation projects.

This paper describes the requirements collected and how they may be implemented in practical applications. We propose a model for representing annotation data and implementing annotation systems. In particular, the model introduces a separate entity database that maintains information about the entities referenced from annotations and can point to additional information about these entities such as Linked Open Data resources.

The result of this work is the annotation model and a collection of requirements that can be applied in different configurations depending on particular use cases and can provide a basis for developing and improving annotation tools.

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

1 Introduction

Digital documents enable data mining and text processing methods that help us discover information patterns and extract new knowledge from these documents. One of the ways to improve document analysis is to identify the facts, objects and other useful information mentioned in documents. There is a substantial amount of previous research done in Named Entity Recognition to mark up mentions of objects of selected types in natural language text documents (Atdağ, S., Vincent 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, might be annotate mentions of Plants, Insects and Mammals in their research documents.

This is particularly true in the cultural heritage domain where libraries, museums and other organisations have large collections of documents spanning from modern works to scans and transcriptions of ancient documents written in old languages and dialects. Various researchers may be interested in recording different kinds of infor-

mation contained in these documents and marking up important areas of text, identifying mentions of named entities and recording other knowledge included in or related to their document collection.

Additionally, it is often necessary to uniquely identify the objects mentioned in the text regardless of the literal form used to refer to them. For example, names of a tree in English and in Latin might still reference the same exact plant. Likewise, a historical text may contain placenames that refer to a location that has a different modern name.

As a proof-of-concept we have selected two datasets – full-text document collections from cultural heritage field provided by the National Library of Latvia – and identified annotation requirements based on these datasets. A web-based annotation tool is under development that will support these annotation requirements in a shared environment.

2 Datasets and Motivation

This research was initiated by exploring the annotation needs of two datasets at the National Library of Latvia (NLL) and realising that the available annotation tools do not fully satisfy these needs. The datasets we used are:

Correspondence (letters) from late 19th century between two of the most famous Latvian poets: Rainis and Aspazija. These letters were available both as images (hand-written text) and as transcripts with additional comments by literature experts.

They contain many references to persons, places, dates, literary works and other entities. Most of these can be uniquely identified and linked to existing authoritative data. The letters have already been extensively analysed as individual documents, but not as a whole document collection. This presents a possibility to discover new, previously unknown facts about both the poets themselves and about other objects mentioned.

The Linked Digital Collection "Rainis and Aspazija" pilot project was created as a first step towards exploring how this material could be enriched and explored using text annotations[1]. Poets' correspondence was annotated with references to the entities mentioned in these letters resulting in a network of links between letters and the entities mentioned in them (Bojārs, 2016).

Parliamentary transcripts that document the first four parliamentary terms in Latvian history (1922-1934). These transcripts are available at NLL both as paper documents and as their digitized versions (National Library of Latvia, 2017). They are more structured than personal correspondence: each parliamentary session is represented as a separate chapter that begins with a table of contents for that particular session followed by session transcript. Every speaker is usually uniquely identified by their name followed by party affiliation or role in this particular case. Transcripts are particularly rich in mentions of persons, places, projects, legal acts and organizations.

[1] The pilot project can be found online at http://runa.lnb.lv .

They also contain information about events such as MPs' trips or vacations, and mentions of other parliamentary debate activities such as interjections or remarks.

The "Rainis and Aspazija" pilot project highlighted issues with available annotation approaches and tools, and acted as valuable input for the text annotation requirements described in this paper. These issues made us go back to the "drawing board" and look at what would be the requirements that a good annotation tool should have in order to satisfy cultural heritage use cases such as the two cases described above.

Datasets were analysed to identify what kinds of annotations would be necessary to fully describe their content and semantics, and what functionality would be needed to create them. This analysis was done together with researchers familiar with the datasets. They described all the information that they would want to annotate these datasets with and how they would want to do it. In the case of the "Rainis and Aspazija" project the researchers had already annotated a part of the dataset and had practical experience to learn from.

We collected and systematised these requirements, generalising where necessary, in order to adapt to other potential use cases for semantic annotation of cultural heritage content. The result is a collection of requirements described in the next section.

3 Annotation Requirements

The following concepts are used in the collection of requirements described in this chapter:

- *Documents* contain the textual content to be annotated;
- *Text fragments* or segments are parts of documents (such as words or sentences) that annotations are attached to;
- *Annotations* attach some information to the text or its fragments;
- *Annotation classes* distinguish between different types of annotations. Annotations of different classes may have their own specific properties;
- *Entities* are distinct, identifiable objects mentioned in the text. Annotations may include references to the relevant Entities;
- *Entity classes* are used to distinguish between different types of entities (e.g. Person or Location);
- the *Entity database* manages all information related to entities.

Text annotation tasks may include various annotation scenarios and use cases 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. a mention of an Event that is described by text fragments containing information about its Date, Location and Participants).

The rest of this section lists the main requirements identified, divided into four groups of requirements – Annotations, Entities, Annotation process and Interoperability.

Annotations:

- Each annotation has an *annotation class* that distinguishes different types of annotations from one another. [A1]
- An *Annotation* may contain a unique reference to the *entity* mentioned in the text. [A2]
- *Standard, predefined annotation classes*. The system must allow administrators to predefine annotation classes such as the common classes used in Named Entity Recognition (NER): Persons, Places, Organizations, Dates, etc. [A3]
- *Users-defined annotation classes.* Besides the standard classes it is often necessary to identify non-standard objects such as Political parties in parliamentary transcripts. These classes may be more specific that the standard classes. Users should be able to define new annotation classes as necessary. [A4]
- *Annotations can have properties.* Because the same object can be mentioned in text many times in different contexts, it is important to be able describe the context for each annotation in its properties. For example, the same person might be representing two different parties in the entire collection of parliamentary transcripts. The party affiliation in this case might be considered a property of a Politician annotation class. [A5]
- *Annotations can reference other annotations*. A typical example here would be an annotation of a class Vacation that would point to one Person annotation and two Date annotations (start and finish of a vacation). This requires more a complex annotation structure where one annotation may reference other annotations. [A6]
- *Annotations have technical metadata.* It is important to identify, who and when created the annotation, what is the visibility of the annotation, etc. [A7]
- *Annotations should have an addressing mechanism that is resilient to changes in the text.* Some texts may need to be corrected after the annotation process has already begun (e.g. misspelled words can be corrected and missing words might be added). In this case the positions of all annotations in the text must be recovered to still point to correct text fragments. [A8]
- *Annotations can have trustworthiness probabilities.* A user might sometimes be unsure whether he has correctly identified the object, the annotation class or even

the text boundaries of the annotation. Such probabilities have already been suggested in Text Encoding Initiative format (TEI Consortium, 2016). [A9]

An important difference from other models is that annotations can have properties (A5) and may reference one another (A6). As a result, annotated documents become local knowledge bases that describe the knowledge contained in the document. While this type of information could potentially be described in the background knowledge base (in our model it is the Entity database) we should distinguish between local, context-dependent knowledge of the annotated document which annotates exactly what is said in the document and the universal, commonly-accepted knowledge that is considered true and important enough to be stored at the global level in the Entity database.

Thus, the Entity database (E2) contains global information about entities while the information stored in annotations is local to the project or document. At some point this local annotation knowledge may be evaluated and determined to be universal and important enough to be "lifted up" into the Entity database.

Entities:

- The objects mentioned in annotations should be uniquely identifiable by linking them to *globally unique entities.* One way to achieve this is to create an Entity database for storing all entity information. We chose this approach in order to maintain all the relevant information in the annotation system. [E1]
- *Entity database* – the system must maintain information about the entities mentioned in the text. This information includes the names (literal forms) of the entity, links to the same entity in other sources and comments. [E2]
- Each entity has an *entity class* that describes what kind of entity it is (e.g. a Person). Unlike annotation classes that describe information in the local, document context, entity classes are a global, higher-level classification of the entity that applies regardless of how it is mentioned in annotations. [E3]
- *Multilingual entity names.* The names of entities can be represented in several different languages. The system must keep track of what language each name is in. [E4]
- Each entity has a primary name (the *standard form*) that is used by the application when referring to it. [E5]
- *Automatic creation of entity records.* Existing authoritative data sources (e.g. a library information system) may already have rich information about required entities. When adding a new entity to the database it must be possible to locate it in relevant databases and import its information automatically. [E6]

The Entity database is necessary for storing all important information related to entities. Its records may contain references to other sources, including Linked Data sources such as VIAF or DBPedia (Lehmann et al., 2015). While it is possible to use multiple Entity databases (provided that entities are referenced using globally unique URIs), for practical purposes we will assume that the annotation system has one global Entity database.

Annotation process:

- *Several users can create annotations in the same document.* The larger the collection of text documents, the greater a probability that several users will be involved in an annotation process. This in turn means that actions of different users must be monitored so that not to create contradictory annotations. [P1]
- *There can be private and public annotations.* A user might want to add their own annotations that are not necessarily required for the general purpose of annotating the particular text collection. [P2]
- *User interface requirements.* The annotation process requires various functionality from the user interface such as a good search interface for both annotations and entities. [P3]

Interoperability:

- *Linked Data*. The information in the annotation system should be accessible in a machine-readable form as Linked Data. A minimum requirement is to have all the information from the Entity database available as Linked Data. The availability of other information (about annotations and their classes, text fragments, etc.) may differ depending on the use case and on privacy considerations. [I1]
- *Annotation information should be identifiable with a URI.* The annotation itself, the annotation class and the object mentioned may be identified with a globally unique URI. This is a pre-requisite for making information available as Linked Data. [I2]
- *Annotation import / export.* Although annotations can be implemented by storing all of their technical parameters in a database, for data interchange purposes there should also be an open format for storing, importing and exporting annotations. [I3]
- *Preserving original formatting.* The system should be able to preserve the core formatting of the original documents for selected data formats. [I4]

This requirement collection can be adapted based on the needs of each particular use case – the items listed here can be combined as necessary and there may be cases where some are not needed or where new requirements arise. This section listed the main annotation requirements identified but we also came up with additional requirements that are useful but not as important to core annotation tasks (e.g. organising documents into document sets or projects) and are too numerous to list in this paper.

4 Annotation Model

Based on the requirements above we propose the following general annotation model. It is based on three core annotation types:

- **Simple annotations**. There are both named entity annotations that link a text fragment to a single entity in Entity database and even simpler annotations that do

not reference a named entity and have the annotation class and an optional comment;

- **Structural annotations**. Sometimes it is important to mark up a fragment of a text document that has a special meaning within a context of that document. There might be quotes, comments, remarks that are semantically different from the core text. They do not have a corresponding entity but their information could be useful in further text analysis by showing that, for example, the keyword was found in a specific part of a document.

- **Composite annotations**. A combination of several annotations can sometimes be required to represent more complex information. For example, an annotation of class *Business trip* might reference information about this business trip marked up by other simple annotations of classes Person, Date and Place. In this case a composite (complex) annotation is created to represent the *Business trip* event. It contains links to the relevant component (simple) annotations. These links may have different types to represent the role each simple annotation plays in the composite annotation.

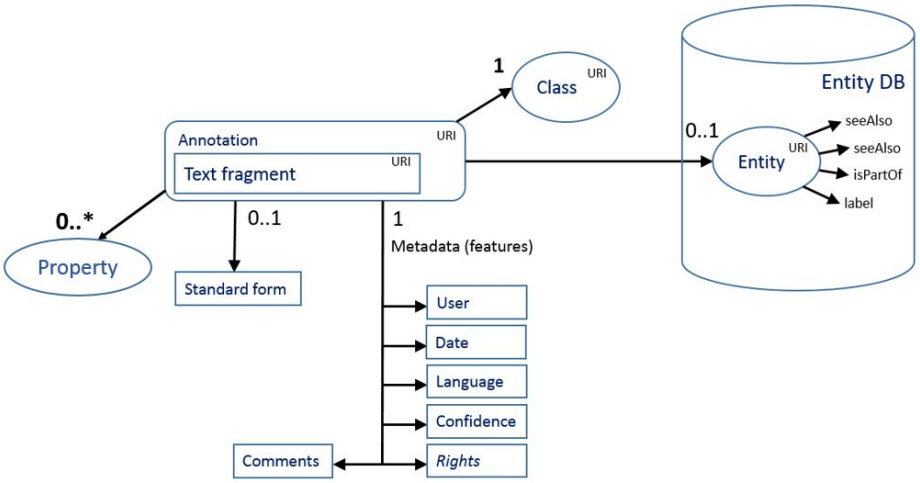

Fig. 1. The model of a simple annotation with a link to the Entity database.

Figure 1 represents the annotation model for the first two types of annotations. These annotations consist of a text fragment reference associated with the annotation class, have technical metadata and may be linked to an entity in the Entity database. Annotations may also have comments. In most cases annotations will reference exactly one entity, however, there might be cases when objects cannot be identified with 100% certainty or the relevant entity has not been identified yet. In that case annotation might omit a link to an entity until it is properly identified.

All annotations have an *annotation class*. It specifies what kind of information is described by the text fragment. It is especially useful in cases where annotation do not

contain entity references (e.g. when the user is pre-annotating text with entity types but is not yet identifying and describing the actual entities mentioned). Annotation classes also have other uses described further in this paper.

The Entity database is not just a simple collection of entities, but can contain links between entities which may be organized in a hierarchical structure or be linked to one another. For example, there might be a link between two Person class objects if the corresponding people are spouses. Libraries may have different entities for Literary works and their Manifestations, and Place type entities may be organised in a hierarchical structure if one place is within another. The Entity database contains information about all the entities referenced in annotations and acts as a background knowledge base about these entities.

Composite annotations are a more complex case as they do not represent a single entity mentioned in the text. As such they do not contain a named entity reference but rather point to one or more existing annotations that are used to mark up the information that describes the thing represented by the composite annotation.

We also propose to distinguish between *annotation classes* and *entity classes*. As described in Section 3, some information about entities can be viewed as universal, commonly-accepted knowledge stored in the Entity database. Annotations may also contain information that is local to the document being annotated and may be context dependent or not necessarily true (i.e. it reflects what is claimed in the document). These two "levels" of knowledge – global and local – may be different in their content and level of detail. As a result, they may require different conceptual schemas (*annotation classes* and *entity classes*). We should also consider *Composite* and *Structural annotations* which do not have corresponding entities and thus cannot be described using the same classes as those used for entities.

Currently a web based annotation tool is under development that will support the annotation model proposed in this paper.

5 Annotation format

After a document collection has been annotated these annotations could be published on the Web or shared with other third-party systems. In order to achieve this an annotation representation format is needed. The format should cover both syntax and semantics. Because annotations are likely to be shared on the Web a widely used open syntax (e.g. XML or JSON) should be used.

We did not find an existing format that covers all the semantics of our annotation model. However, the W3C Web Annotation format (Sanderson et al., 2017) is a good basis which we could tailor to our model. This format uses JSON for syntax and is extensible so that we could include annotation features originally lacking in the Web Annotation Model (WAM). An important part of WAM is content segment selectors which can be directly reused to identify text fragments.

Let us imagine an example of an Annotation mentioning a particular Person that has been identified as *Frīdrihs Vesmanis* and described using an entity database record linking to an entry in VIAF (OCLC, 2016). Our tailored WAM representation for this annotation would be:

```
{
  "@context": "http://www.w3.org/ns/anno.jsonld",
  "id": "http://data.lndb.lv/Saeima/doc_s01/anno01",
  "type": ["Annotation","ObjectAnnotation"],
  "annotation_class": "Person",
  "body": {
    "type": "SpecificResource",
    "source": "http://data.lndb.lv/entity/
Fridrihs_Vesmanis",
    "purpose": "identifying"
  },
  "target": {
    "source": "http://data.lndb.lv/Saeima/doc_s01",
    "selector": {
      "type": "TextPositionSelector",
      "start": 15,
      "end": 27
    }
  }
}
```

We follow the W3C Web Annotation data model (Sanderson et al., 2017) where the annotation is a graph representing a relationship between resources. Typically, an annotation has one *target* element (resource) that identifies the content to be annotated and one *body* element that describes some information about this content – in this case it is the entity the annotation refers to (a URI of its record in the Entity database).

The *id* element contains the URI of the annotation. The URI of the entity referenced from the annotation is located in the *source* element of the annotation *body*.

Text fragments are identified using standard mechanisms provided by WAM. In this case the fragment is identified by text start and end positions. This poses the risk of losing annotation's position in text if changes to text are allowed (A8). One solution is to monitor all changes and to recalculate annotation's position after every change. However, it is not always clear how to maintain the exact extent of the annotation. Another option is to use several independent addressing mechanisms together. WAM helps solve this by allowing multiple segment selectors to be used at once. A good option would be to also preserve the prefix and suffix of the annotation – text fragments right before and after the annotation in question.

The extensions to the WAM format that are needed in order to support the annotation requirements identified in this paper include:

- The *type* element defines the Annotation type. In addition to the value “Annotation” (defined by WAM) it also needs to represent the core annotation type in our model (simple, structural or composite annotation). This is done using one of the

following values: “ObjectAnnotation”, “StructuralAnnotation” and “CompositeAnnotation”.
- The *Annotation class* (A1) is represented by the *annotation_class* attribute.
- Additional information needed for representing annotation properties and references to other annotations is stored in the *body* element using a new *attributes* element.

The following example shows the representation of a Composite annotation representing information about a parliamentary record of MP's vacation. It references other simple annotations that mark up text fragments that describe the MP (Person) who is going on a vacation and relevant vacation start and end dates. References to these annotations are stored in the *attributes* element described above along with information about the roles which they play in the Composite annotation.

```
{
  "@context": "http://www.w3.org/ns/anno.jsonld",
  "id": "http://data.lndb.lv/Saeima/doc_s02/anno_complex01",
  "type": ["Annotation", "CompositeAnnotation"],
  "annotation_class": "Vacation",
  "body": {
    "type": "CompositeAnnotationBody",
    "attributes": {
       "person": "http://data.lndb.lv/Saeima/doc_s02/anno01",
       "date_start":
"http://data.lndb.lv/Saeima/doc_s02/anno02",
       "date_end": "http://data.lndb.lv/Saeima/doc_s02/anno03"
    },
    "purpose": "describing"
  }
}
```

In some, more specific cases further WAM adaptations may be needed such as adding a *probability* attribute to represent the certainty of the annotation.

6 Conclusions

This paper defines a set of requirements and a model for text annotation that enable use cases that often appear in cultural heritage annotation projects. These requirements are based on the analysis of the needs of existing and planned cultural heritage document annotation projects at the National Library of Latvia.

Annotations are divided into three core types: simple, structural and composite annotations, where simple annotations can be used for marking up mentions of named entities while composite annotations may describe more complex information such as events where parts of the information being annotated are described by other simple annotations. An important part of the model is the Entity database that contains all

relevant information about the entities mentioned in annotations, including links to external sources describing these entities (such as Linked Open Data resources).

Although the annotation model and requirements were created by analysing cultural heritage use cases, the proposed solution could also be applied to other fields that deal with text documents such as research papers, news articles or technical documentation.

Acknowledgments. This work was supported by the European Regional Development Fund under the project "IT Competence Centre" (1.2.1.1/16/A/007) and is part of the individual research project no. 2.1. "Semantic annotation of textual data in web environment for related data sets".

References

1. Atdağ, S., & Labatut, V. (2013). A comparison of named entity recognition tools applied to biographical texts. In the 2nd International Conference on Systems and Computer Science (ICSCS), 2013, pp. 228-233. IEEE.
2. 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), pp.21-26. http://ceur-ws.org/Vol-1608/paper-04.pdf
3. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., & Bizer, C. (2015). DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2), 167-195.
4. National Library of Latvia (2017). Saeimas stenogrammas. Available at: http://periodika.lv/#periodical;id=120443059500088665005414615187708183054 [Accessed July 17, 2017].
5. OCLC (2016). VIAF: The Virtual International Authority File. Available at: https://viaf.org/ [Accessed May 29, 2017].
6. Sanderson, R., Ciccarese, P., & Young, B. (2017). Web Annotation Data Model. W3C Recommendation. 23 February, 2017. Available at: https://www.w3.org/TR/annotation-model/
7. TEI Consortium (2016). Guidelines for Electronic Text Encoding and Interchange. Available at: http://www.tei-c.org/Guidelines/P5/ [Accessed May 29, 2017].