A Flexible Representation of Heterogeneous Annotation Data

Richard Johansson, Alessandro Moschitti

Department of Information Engineering and Computer Science
University of Trento
Trento, Italy
{johansson, moschitti}@disi.unitn.it

Abstract

This paper describes a new flexible representation for the annotation of complex structures of metadata over heterogeneous data collections containing text and other types of media such as images or audio files. We argue that existing frameworks are not suitable for this purpose, most importantly because they do not easily generalize to multi-document and multimodal corpora, and because they often require the use of particular software frameworks. In the paper, we define a data model to represent such structured data over multimodal collections. Furthermore, we define a surface realization of the data structure as a simple and readable XML format. We present two examples of annotation tasks to illustrate how the representation and format work for complex structures involving multimodal annotation and cross-document links. The representation described here has been used in a large-scale project focusing on the annotation of a wide range of information – from low-level features to high-level semantics – in a multimodal data collection containing both text and images.

1. Introduction

Annotated data resources, and the infrastructure to process them, form the backbone of the language technology research enterprise. The development of standards that describe how to conceptually represent annotations and how to serialize them as files is thus an important subarea in language resource research. For textual data, the most prominent representation (and corresponding serialization) is undoubtedly TEI (Sperberg-McQueen, 2002), which has a decades-long history in the humanities but has been less influential in the language technology community. A recent alternative, explicitly intended for language processing technology applications, is UIMA (Ferrucci and Lally, 2004), whose representation is tightly coupled to a Java-based software architecture. A similar example is the representation used by GATE (Cunningham et al., 2002). In addition to the generic formats mentioned here, there are also specialized formats; one well-designed representation worth mentioning is the TigerXML format used to represent syntactic trees in treebanks (Brants et al., 2002).

We envision that with the increasing availability of multimodal web data, there is a need for a representation that is explicitly designed for heterogeneous annotated data. We need to be able to annotate metadata on top of not only text but media of various types such as images or speech. This makes it difficult to use existing frameworks that are exclusively text-oriented.

Another shortcoming of previously proposed frameworks and file formats is that they are *centralized* to a large extent. In order to facilitate large-scale, parallel processing, the representation should be decentralized and modular, so that processors can load only the parts that they need.

In this paper, we present a new data representation for annotated resources. It has been designed to meet the following requirements:

- modularity – a processing module should only need to load the data it will use,
- redundancy – should be able to handle output from several modules of the same type (such as a collection of POS taggers) or annotated by different human annotators on top of the same base,
- flexibility – a meta-format that can be specialized to represent structured data of various types,
- stand-alone – no lock-in to a particular software framework, allowing processors to be implemented in multiple programming languages and employ the most efficient internal representation for the particular task at hand,
- cross-document – should not be restricted to annotating data in a single document
- multimodality – annotations can be grounded not only in text but also in other types of media, such as images.

The proposed data representation model has been applied practically in the LivingKnowledge research project, which focuses on annotating semantic information in large collections of web-crawled text and image data.

2. Conceptual Organization

Conceptually, an annotated collection of data consists of

- unstructured data such as raw text and images; we refer to this as the *ground data*,
- annotation data structures.

Similar to other annotation models, we organize the annotated data as an annotation graph (Bird and Liberman, 2001). The nodes of the graph are the annotation entities, that is the pieces of data we annotate on top of the raw data. Figure 1 shows an example of a data collection and the representation of its annotated data.

2.1. Definitions

We now formally define the concepts used above. The most important building block in the representation is the annotation entity, which we now define.

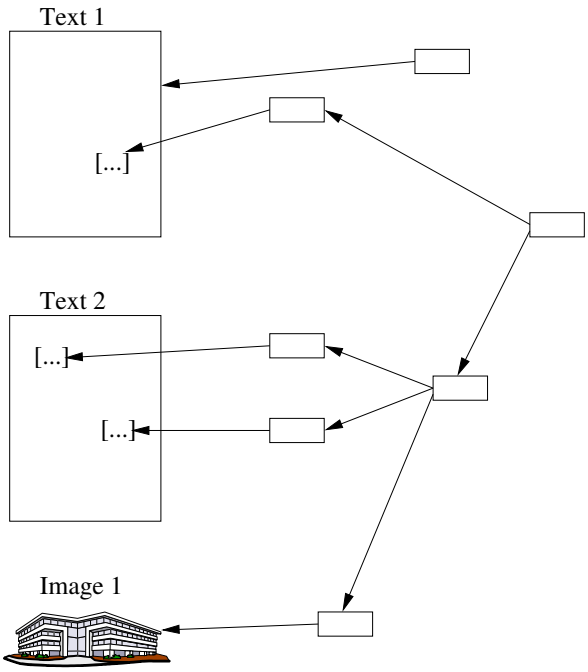

Figure 1: Example of a graph of annotation entities over a data collection.

Definition. An *annotation entity* is a tuple (D, R) where D is a piece of metadata – the *payload* – and R, the *referent*, belongs to one of the four following types:

- A file of ground data. A prototypical example of this would be a category label on a document or image.
- A subregion of a file of ground data. This is frequently used in tokenization of text.
- A set of annotation entities.
- A pair of annotation entities. This represents a binary relation, such as in a syntactic dependency graph.

To facilitate modularization and conceptual organization, annotation entities are grouped into *annotation layers*. An annotation layer is simply defined as a named list of entities.

As a basic example, we may create an annotation layer consisting of token entities, where each token points to a subregion of a text file. A set of tokens may be grouped into a sentence, which may then form part of a paragraph.

2.2. Multimodality

A central purpose on the format is that it can be used in a project where one of the main goals is to explore the interaction between text and images. Therefore, we need to distinguish annotations applied to an image *in isolation* or *in context*, i.e. as it appears in a document, surrounded by text and possibly also other images. As an example of context-independent image data, we may annotate whether or not the image contains a human face. Context-dependent information typically represents the *function* of the image in a document, such as whether or not the image is used to clarify an exposition.

For an image considered in isolation, our annotation tools simply create entities pointing directly to the image file. When we consider an image in the context of a document, we create a special annotation layer for that document, in which the entities point to the image file and to the textual position. Annotations that refer to the image in the document then refer down to this layer.

3. Surface Representation

The data structures described in the previous section are serialized using an XML format. As explicitly required, this format is modular; the annotation layers referring to a certain document or image may be spread over several files. We specify two types of files:

Raw text files. Although the raw text is formally treated as a ground file type like any other, we use a special format to store the raw text. This is useful since it may be interesting to store details about the extraction of the raw text from its original source, which may be a partly structural format such as HTML or PDF.

Annotation files. In this type of file, one or more annotation layers are stored. Every layer defines a *provision* identifier describing the type of its data.

In the headers of these files, we store information about the processing of the data. This is important for traceability. Important examples of annotation metadata that would be present in most cases include the original source document and an identifier of the processing tool that output the annotation. This metadata can be used to separate the annotations provided by different tools or human annotators.

3.1. Serialization of References

References are serialized as URIs. Although the use of full URIs may result in verbosity, and our format is intended to be used with large data collections, XML entities can be used to make the files less wasteful. For references to annotation entities, we employ fragment URIs, where the fragment part consists of the identifier of the referent entity. Although all sets of entities can be specified explicitly, we provide a special syntactic construction to describe *ranges* of adjacent entities in a layer. This is often used in text-based annotation, for instance when tokens are grouped into phrases or sentences.

As described in the previous section, we provide two ways to ground the annotation in the raw data: pointing to a full file and pointing to a subregion. Currently, subregions are only defined for text, where they take the form of start–end ranges of characters.

3.2. Serialization of Payload

Since we cannot know what types of data payload that will be used in a particular application, we impose no restrictions on the structure of the payload. The format thus allows any text or XML structure to be stored as payload, as long as it is properly escaped not to break XML syntax. In many cases, the payload will be an atomic label such as a named entity tag (`PERSON`, `COUNTRY` etc), but in more complex analysis tasks it may be necessary to store structured data in the payload.

4. Examples

To illustrate how the representation and its serialization work in practice, we give two examples: first an example on multi-document coreference annotation to illustrate cross-document linking of entites, then an example involving images.

4.1. Cross-document Coreference Relations

As an example of multi-document annotation, consider the task of annotating coreference relations between named entities. Figure 2 shows how this kind of annotation can be organized. For every document, there is a token layer pointing directly into the text, and a named-entity layer pointing to the tokens. For every coreference equivalence class, we create an *anchor* pointing to the first mention of the named entity, and finally coreference links such as identity (`ID`) or subset (`SS`) connecting individual mentions to the anchor representing the whole coreference chain. As opposed to the token and named entity layers, the coreference layers cover the whole collection, not just single documents. The graph of annotations over the collection can then be serialized to XML. Figure 3 shows the serialization of the coreference layers.

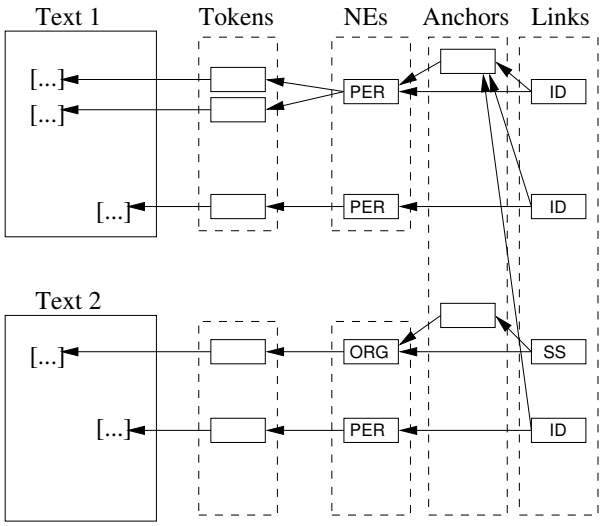

Figure 2: Example of multi-document coreference annotation.

```
<layers>
  <meta-info>
    <tag name="annotator">COREF</tag>
  </meta-info>

  <layer provides="COREF-anchors">
    <e id="1" on="text1_NE.xml#5"/>
    <e id="2" on="text2_NE.xml#3"/>
    ...
  </layer>

  <layer provides="COREF-links">
    <e id="20" from="#1"
       to="text1_NE.xml#5">ID</e>
    <e id="21" from="#1"
       to="text1_NE.xml#8">ID</e>
    <e id="22" from="#1"
       to="text2_NE.xml#6">ID</e>
    ...
    <e id="31" from="#2"
       to="text2_NE.xml#2">SS</e>
    ...
  </layer>
</layers
```

Figure 3: Example of an XML serialization of coreference layers.

4.2. Annotation on Images

As an example of annotation of information derived from the interplay between documents and images, we show how to annotate the fact that persons mentioned in a document appear in an image referenced by that document. Figure 4 shows the organization of this data structure. We first create an annotation layer to represent the use of an image in a particular textual context. Then, we create links that connect these entities to the named entities occurring in the text. Figure 5 shows the serialization of these structures.

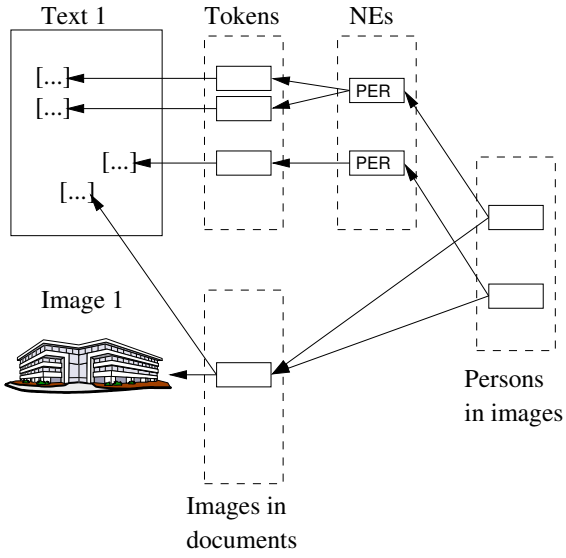

Figure 4: Example of annotation on images.

5. Discussion

We have argued that a new flexible representation is needed to annotate structures of metadata over heterogeneous data collections containing both text and images. We suggested that existing frameworks are not suitable for this purpose. Consequently, we outlined a new representation and a corresponding surface realization, and gave examples of their use.

Since we do not want to create a lock-in situation, we have defined a format and stayed short of mandating the use of a software platform or programming language. Instead, we encourage implementers to employ the most efficient *internal* representation for the specific case. However, in the

```
<layers>
  <meta-info>
   <tag name="annotator">ImageInfo</tag>
  </meta-info>

  <layer provides="ImagesInDocs">
    <e id="1" from="text1.xml#105"
              to="image1.png"/>
    ...
  </layer>

  <layer provides="PersonsInImages">
    <e id="10" from="text1_NE.xml#5"
               to="#1"/>
    <e id="11" from="text1_NE.xml#8"
               to="#1"/>
    ...
  </layer>
</layers
```

Figure 5: Example of an XML serialization of image annotation layers.

project where the format has been applied, a wrapper into UIMA has also been developed for subsets of the data. The easiness of adaptation depends on the type of collection – with many cross-document links, this becomes infeasible. It should be noted that the internal representation used in a particular piece of software making use of the format does not have to be isomorphic to the structure encoded by the files. Redundancy may need to be introduced internally to improve efficiency. For instance, since the format represents only downward links – from higher abstractions down to the grounding material – an application that finds all annotations on a given image or named entity may need to reverse the annotation links internally.

6. Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 231126: LivingKnowledge – Facts, Opinions and Bias in Time, and from Trustworthy Eternal Systems via Evolving Software, Data and Knowledge (EternalS, project number FP7 247758).

7. References

Steven Bird and Mark Liberman. 2001. A formal framework for linguistic annotation. *Speech Communication*, 33(1,2):23–60.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theory*, pages 24–41, Sozopol, Bulgaria.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

David Ferrucci and Adam Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348.

C. Michael Sperberg-McQueen. 2002. *Guidelines for Electronic Text Encoding and Interchange*. University of Virginia Press.