Representing Linguistic Corpora and Their Annotations

Nancy Ide*, Laurent Romary**

*Department of Computer Science
Vassar College
Poughkeepsie, New York 12604-0520 USA
ide@cs.vassar.edu

**Equipe Langue et Dialogue
LORIA/CNRS
Vandoeuvre-lès-Nancy, FRANCE
romary@loria.fr

Abstract

A Linguistic Annotation Framework (LAF) is being developed within the International Standards Organization Technical Committee 37 Sub-committee on Language Resource Management (ISO TC37 SC4). LAF is intended to provide a standardized means to represent linguistic data and its annotations that is defined broadly enough to accommodate all types of linguistic annotations, and at the same time provide means to represent precise and potentially complex linguistic information. The general principles informing the design of LAF have been previously reported (Ide and Romary, 2003; Ide and Romary, 2004a). This paper describes some of the more technical aspects of the LAF design that have been addressed in the process of finalizing the specifications for the standard.

1. Introduction

A Linguistic Annotation Framework (LAF) is being developed within the International Standards Organization Technical Committee 37 Sub-committee on Language Resource Management (ISO TC37 SC4)[1]. LAF is intended to provide a standardized means to represent linguistic data and its annotations that is defined broadly enough to accommodate all types of linguistic annotations, and at the same time provide means to represent precise and potentially complex linguistic information. The general principles informing the design of LAF have been previously reported (Ide and Romary, 2003; Ide and Romary, 2004a). This paper describes some of the more technical aspects of the LAF design that have been addressed in the process of finalizing the specifications for the standard.

2. LAF Design

The Linguistic Annotation Framework is designed based on the following requirements:

- LAF must enable users to represent their data and annotations in a variety of formats of their own choosing.
- LAF must accommodate all varieties of annotation and data (including, e.g., time-stamped speech, streamed data, etc.).
- LAF must be easy to use so that the community will adopt it.

To meet these requirements, we have defined an abstract data model for annotations, such that annotation formats conforming to the model—regardless of differences in their superficial representation—are trivially mappable to one another. The model is instantiated by a "dump" format that is intended to function in the same way as an interlingua functions for machine translation--i.e., as an abstract representation of universal concepts into and out of which realizations in different languages are mapped for the purposes of translation. Here, the different languages are different user-defined formats.

The overall LAF design is based on a few straightforward principles:

- **Separation of data and annotations.** Language data is regarded as "read-only" and contains no annotations. All annotations are contained in stand-off documents linked to the primary data. Note that we define primary data as the source data obtained by the user, which may include markup (e.g. HTML tags, time stamps, etc.), but to which no linguistic annotations that will be rendered into the dump format have been superimposed.[2] Treating the primary data as read-only avoids maintenance problems associated with stand-off approaches, where modification of the data may cause links into it to break.
- **Separation of user annotation formats and the exchange ("dump") format.** Users may use any format for annotations, including not only XML but also formats such as LISP-like structures or tab-delimited information, etc. The only requirement is that the information in the user's annotation format is automatically mappable to the feature structure-based data model instantiated by the dump format.
- **Separation of structure and content in the dump format**. In many annotation schemes, content and structure are not clearly differentiated. In such cases, structural relations among parts of the annotation content may be ambiguous. The most obvious example is LISP-like formats, which use parentheses

[1] http://www.tc37sc4.org

[2] We recognize that the line between source data and annotations is a highly debatable matter; our definition here is driven by the practical concerns of dealing with web and legacy data. We believe that in most cases, there is a common sense distinction between data and annotations that can be applied.

to group information, with no indication of whether the group represents an inclusive list, a prioritized list, a set of alternatives, etc. The only way to determine which applies is to examine the data; if, for instance, the list describes syntactic frames or part of speech for a given lexical item, it is probably a set of alternatives, but human knowledge is required to decide this and program a script to treat it appropriately. LAF requires that all annotation information included in the original format be made explicit in the dump format representation; this way, fully automatic transduction from the dump format representation or other user formats is ensured.

The dump format represents an annotation as a directed graph referencing *n*-dimensional regions of primary data as well as other annotations. In the primary data, the nodes of the graph are virtual, located between each "character" in the primary data, where a character is defined to be a contiguous byte sequence of a specified length.[3] When an annotation references another annotation document rather than primary data, the nodes are the edges within that document that have been defined over the primary data or other annotation documents. That is, given a graph, *G*, over primary data, we create an *edge graph G'* whose nodes can themselves be annotated, thereby allowing for edges between the edges of the original graph *G*. Edges are labeled with feature structures containing the annotation content relevant to the data identified by the edge.

The dump format is instantiated in XML. ISO TC37 SC4 has collaborated with the Text Encoding Initiative (TEI) Consortium to adapt and revise the TEI's specifications for representing feature structures in XML[4]. The ISO/TEI specifications implement the full power of feature structures and define inheritance, unification, and subsumption mechanisms over the structures, thus enabling the representation of linguistic information at any level of complexity. The specifications also provide a concise format for representing simple feature-value pairs, which suffices to represent many annotations.

It is important to note that in principle, the dump format places no restrictions on annotation content (i.e., the categories and values in an annotation); annotation content is effectively user-defined, taken directly from the user's original annotation. However, it is obvious that harmonization of content categories is a critical next step toward standardizing annotations. LAF is addressing this far more controversial and problematic issue separately. Two major activities within SC4 are aimed at harmonization of annotation content: (1) definition of user annotation formats for different annotation levels[5], and (2) creation of a Data Category Registry (DCR) containing pre-defined data elements and schemas that can be used directly in annotations (Ide and Romary, 2004b). The DCR includes atomic data category (both category names and values) that may be referenced directly in user annotations, or to which a mapping from user –defined categories can be included in the dump format representation. In addition, feature structure libraries that can be referenced directly in both user and dump format annotations are under development.

3. Dump Format Design

In principle, users will never deal directly with, or even see, the dump format, and therefore the primary concerns in designing the dump format representation are to

- maximize processing efficiency and consistency;
- ensure that processing is unambiguous;
- ensure that the mapping from user formats is not overly complex.

Fulfillment of these requirements has repercussions for users, because it demands that certain information is explicitly provided in their representations or made explicit via the mapping from the user-defined format to the dump format.

The following outlines some of the technical aspects of the dump format instantiation.

3.1. Segmentation

As noted earlier, in the dump format, primary data contains no annotations and is regarded as "read-only". Therefore, LAF insists on the existence of a "segmentation" annotation document for the primary data that identifies contiguous sequences of characters (bytes) comprising a logical unit. *Primary segmentation documents* contain no annotations; they serve solely to identify the base edge set for an annotation or several layers of annotation. Multiple segmentation documents can be defined over the primary data, and multiple annotation documents may refer to the same segmentation document.

For text, the most common primary segmentation is the token, over which, for example, word forms (which may or may not consist of contiguous tokens) may be defined for the purposes of morpho-syntactic annotation. However, edges can be defined over any span of contiguous primary data, regardless of its length.

Any annotation document can be treated as a *virtual segmentation document* by another annotation. In both primary and virtual segmentation documents, annotations may refer to (1) an edge defined in the segmentation document, in which case the annotation provides information associated with the pre-defined edge; or (2) an *edge graph* consisting of two or more edges in the segmentation document. The latter enables referencing discontiguous entities.

Every annotation document is associated via information in its header with the document that provides the segmentation relevant to that annotation.

3.2. Separate Annotation Layers

LAF defines a fixed set of annotation layers for linguistic annotation. This dictates that to render user-defined annotation formats into the dump format,

[3] For text, the default is UTF-16.

[4] See ISO TC37 SC4 document N188, Feature Structures-Part 1: Feature Structure Representation (2005-10-01), available at http://www.tc37sc4.org/

[5] Draft documents and working papers for the various areas, including morpho-syntactic annotation (ISO TC37 SC4 document N225), syntactic annotation (ISO TC37 SC4 document N244), word segmentation (ISO TC37 SC4 document 233), etc. are available at http://www.tc37sc4.org/.

annotations that may exist in a single document in their representation are separated.

3.3. Sub-component relations

Many annotation types—in particular, syntactic annotations representing phrase-structure—are represented as hierarchical structures over primary data. Given that the dump format is instantiated in XML, it is possible to represent these relations by exploiting the embedding of XML elements. If this representation were used in the dump format, it would be necessary to restrict the semantics of the structure to a single, unambiguous meaning (embedded nodes are sub-components in an ordered sequence). Alternatively, the representation of the annotation structure can, in XML terms, be completely flat, with no embedding of XML elements, and with all sub-component relations represented explicitly by labeled references.

Consistency dictates that the flat structure be used, so that any processor need handle only a single representation of structural relations. However, given that the dump format is instantiated in XML, it is inevitable that an XML parser will be used to interpret it, in which case it is more efficient to exploit the structural information already available to the processor (provided that element nesting is restricted to a single meaning). Although the final recommendation is not fixed, the exploitation of XML embedding to represent constituency relations is currently the favored alternative.

3.4. Overlapping Hierarchies

LAF's insistence on separation of annotations into different stand-off documents avoids the overlapping hierarchy problem in dump format documents. However, the problem must be addressed for users who will combine annotation levels, or different annotations of the same type, for their own use. Several solutions to the overlapping hierarchy problem have been proposed, the most promising of which is CLIX (DeRose, 2005), a solution that is seeing increasing use as a result of its inclusion in OSIS (Durusau and DeRose, 2003), a standard XML schema for Biblical and related materials.

In CLIX, overlaps are handled by the introduction of two attributes, *sID* and *eID*, and by allowing empty content on potentially overlapping XML elements. So, for example, overlap created by merging overlapping sentence and quote annotations into a single document, such as

```
<q id="foo">...<s id="bar">...</q>...</s>
```

is rendered in CLIX as

```
<q sID="foo"/>...<s sID="bar"/>...<q
eID="foo"/>...<s eID="bar"/>
```

To constrain which elements may cross another element's boundaries, "milestoned" elements—i.e., those which overlap and therefore appear in the merged document as empty elements with *sID* and *eID* attributes—appear in a separate XML namespace. This makes them distinct to an XML validator, and thus (for example) `milestones:q` can be allowed only in certain contexts, rather than all and only the contexts where a non-overlapping `<q>` is allowed.

CLIX has the advantage of simplicity, and a CLIX checker has been implemented on top of SAXON, a standard XML parser (DeRose, 2005). XSLT applications have also been developed that handle CLIX.

4. Example

Figure 1 provides a graphical representation of primary data, edges defined over the primary data in a primary segmentation document, and an annotation that references edges identified in the primary segmentation document.

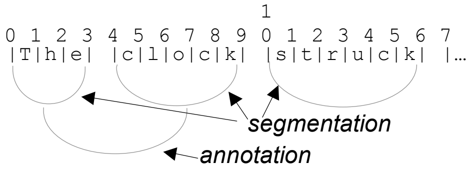

Figure 1. Segmentation and annotation in the LAF dump format

As noted above, the primary segmentation document references virtual nodes located between each character in the primary data. The segmentation document in the dump format for the scenario in Figure 1 would therefore include the following XML:

```
<!-- edges over primary data -->
<edge id="e1" from="0" to="3"/>
<edge id="e2" from="4" to="9"/>
<edge id="e2" from="10" to="16"/>
```

The *from* and *to* attributes specify the start and end nodes for the edge; the actual byte offset within the data is a function of the character encoding.

The following shows a fragment of an annotation document for morpho-syntax that provides information associated with one of the edges defined in the primary segmentation document[6]:

```
<edge id="t2" ref="e2">
   <fs type="token">
      <f name="lemma" sVal="clock"/>
      <f name="pos" sVal="NN"/>
   </fs>
</edge>
```

The `<edge>` element refers to the pre-defined edge with id *e2* in the primary segmentation document. The feature structure included within the edge element provides the annotation: two simple feature/value pairs specifying lemma and part of speech.

When two or more edges are referenced from a single annotation, their id's are provided in an ordered list as the value of the *targets* attribute on the relevant `<edge>` element. The following example shows an annotation for the noun phrase "the clock":

[6] Note that the use of feature structure libraries enables an even more concise XML format for representing feature structures than is shown here.

```
 <!-- edge graph for "The" and "clock" -->
<edge id="np1" targets="t1 t2">
   <fs type="NP">
      <f name="number" sVal="singular"/>
   </fs>
</edge>
```

The effect of this notation is to concatenate the edges referenced in the *targets* attribute and create a new edge from the start node of the first to the end node of the last in the list, which is associated with the annotation.

As many annotations as desired, of either the same type or different types, can reference the same segmentation document or be layered over lower-level annotations. For example, additional morpho-syntactic annotations could reference the primary segmentation document above; a co-reference annotation could reference the NP annotation; and a syntactic annotation could reference the same NP annotation. Figure 2 shows how several segmentation and annotation documents can be layered and inter-leaved over primary data.

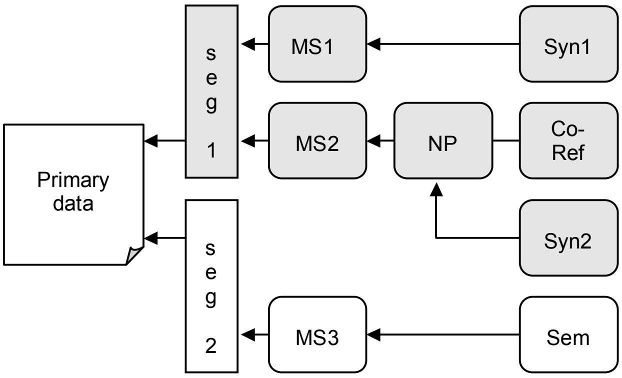

Figure 2. Dump format scenario with two segmentations and multiple annotations

5. Summary

The Linguistic Annotation framework has been under development for several years. As the standard has developed, it has presented to the community in order to receive feedback to inform LAF's continued development. Because harmonization of resources developed for different languages and annotation types is a necessary goal, community input and ultimate acceptance is critical. Comments and input continue to be solicited, both informally and through official ISO channels.

Technical considerations underlying the development of the dump format have been presented here, but it is important to emphasize that the dump format should, in principle, be invisible to most users. Precise specifications of the dump format will be available for the purposes of mapping, but annotators and developers of language resources will continue to work with their own formats. The relevant product of LAF development for all users is the abstract model underlying the dump format and the considerations that have informed its design. The model has been developed based on analysis of annotation schemes of all types and provenance, and is intended to eliminate the difficulties that currently hinder the reuse of language resources. The analysis has shown that these difficulties arise primarily from two sources: implicit information in the annotation that requires human intervention to interpret, and inconsistencies in the means by which annotation content is represented. If these problems are rectified in current annotation schemes via mapping to the dump format, and if they inform the design of new formats, a major step toward resource harmonization will be made.

6. References

DeRose, S. J. (2004). Markup Overlap: A Review and a Horse. *Extreme Markup Languages 2004: Proceedings*. http://www.mulberrytech.com/Extreme/Proceedings/html/2004/DeRose01/EML2004DeRose01.html

Durusau, P., DeRose. S. J. (2003). OSIS: A Users' Guide to the Open Scripture Information Standard. Bible Technologies Group.

Ide, N., Romary, L. (2003). Outline of the International Standard Linguistic Annotation Framework. In *Proceedings of the ACL'03 Workshop on Linguistic Annotation: Getting the Model Right,* Sapporo, pp. 1-5.

Ide, N., Romary, L. (2004a). International Standard for a Linguistic Annotation Framework. *Journal of Natural Language Engineering,* 10:3-4, pp. 211-225.

Ide, N., Romary, L. (2004b). A Registry of Standard Data Categories for Linguistic Annotation. In *Proceedings of the Fourth International Language Resources and Evaluation Conference* (LREC), Lisbon, pp. 135-39.