Outline of the International Standard Linguistic Annotation Framework

Nancy Ide
Dept. of Computer Science
Vassar College
Poughkeepsie,
New York 12604-0520
USA
ide@cs.vassar.edu

Laurent Romary
Equipe Langue et Dialogue
LORIA/INRIA
Vandoeuvre-lès-Nancy
FRANCE
romary@loria.fr

Abstract

This paper describes the outline of a linguistic annotation framework under development by ISO TC37 SC WG1-1. This international standard provides an architecture for the creation, annotation, and manipulation of linguistic resources and processing software. The goal is to provide maximum flexibility for encoders and annotators, while at the same time enabling interchange and re-use of annotated linguistic resources. We describe here the outline of the standard for the purposes of enabling annotators to begin to explore how their schemes may map into the framework.

1 Introduction

Over the past 15-20 years, increasingly large bodies of language resources have been created and annotated by the language engineering community. Certain fundamental representation principles have been widely adopted, such as the use of stand-off annotation, use of XML, etc., and several attempts to provide generalized annotation mechanisms and formats have been developed (e.g., XCES, annotation graphs). However, it remains the case that annotation formats often vary considerably from resource to resource, often to satisfy constraints imposed by particular processing software. The language processing community has recognized that commonality and interoperability are increasingly imperative to enable sharing, merging, and comparison of language resources.

To provide an infra-structure and framework for language resource development and use, the International Organization for Standardization (ISO) has formed a sub-committee (SC4) under Technical Committee 37 (TC37, Terminology and Other Language Resources) devoted to Language Resource Management. The objective of ISO/TC 37/SC 4 is to prepare international standards and guidelines for effective language resource management in applications in the multilingual information society. To this end, the committee is developing principles and methods for creating, coding, processing and managing language resources, such as written corpora, lexical corpora, speech corpora, and dictionary compiling and classification schemes. The focus of the work is on data modeling, markup, data exchange and the evaluation of language resources other than terminologies (which have already been treated in ISO/TC 37). The worldwide use of ISO/TC 37/SC 4 standards should improve information management within industrial, technical and scientific environments, and increase efficiency in computer-supported language communication.

Within ISO/TC 37/SC 4, a working group (WG1-1) has been established to develop a Linguistic Annotation Framework (LAF) that can serve as a basis for harmonizing existing language resources as

well as developing new ones. The overall design of the architecture and the data model that it will instantiate have been described in Ide *et al.,* 2003. In this paper we provide a description of the data model and its instantiations in LAF, in order to enable annotators to begin to explore how their schemes will map into the framework.

2 Terms and definitions

The following terms and definitions are used in the discussion that follows:

Annotation: The process of adding linguistic information to language data ("annotation of a corpus") or the linguistic information itself ("an annotation"), independent of its representation. For example, one may annotate a document for syntax using a LISP-like representation, an XML representation, etc.

Representation: The format in which the annotation is rendered, e.g. XML, LISP, etc. independent of its content. For example, a phrase structure syntactic annotation and a dependency-based annotation may both be represented using XML, even though the annotation information itself is very different.

Types of Annotation*:* We distinguish two fundamental types of annotation activity:

1. *Segmentation:* delimits linguistic elements that appear in the primary data. Including
    - continuous segments (appear contiguously in the primary data)
    - super- and sub-segments, where groups of segments will comprise the parts of a larger segment (e.g., a contiguous word segments typically comprise a sentence segment)
    - discontinuous segments (linked continuous segments)
    - landmarks (e.g. time stamps) that note a point in the primary data

    In current practice, segmental information may or may not appear in the document containing the primary data itself. Documents considered to be *read-only,* for example, might be segmented by specifying byte offsets into the primary document where a given segment begins and ends.
2. *Linguistic annotation:* provides linguistic information about the segments in the primary data, e.g., a morpho-syntactic annotation in which a part of speech and lemma are associated with each segment in the data. Note that the identification of a segment as a word, sentence, noun phrase, etc. also constitutes linguistic annotation. In current practice, when it is possible to do so, segmentation and identification of the linguistic role or properties of that segment are often combined (e.g., syntactic bracketing, or delimiting each word in the document with an XML tag that identifies the segment as a word, sentence, etc.).

Stand-off annotation: Annotations layered over a given primary document and instantiated in a document separate from that containing the primary data. Stand-off annotations refer to specific locations in the primary data, by addressing byte offsets, elements, etc. to which the annotation applies. Multiple stand-off annotation documents for a given type of annotation can refer to the same primary document (e.g., two different part of speech annotations for a given text). There is no requirement that a single XML-compliant document may be created by merging stand-off annotation documents with the primary data; that is, two annotation documents may specify trees over the primary data that contain overlapping hierarchies.

3 LAF overview

LAF development has proceeded by first identifying an abstract *data model* that can formally describe linguistic annotations, distinct from any particular representation (as defined in the previous section). Development of this model has been discussed extensively within the language engineering community and tested on a variety of annotation types (see Ide and Romary, 2001a, 2001b, 2002). The data model forms the core of the framework by serving as the reference point for all annotation representation schemes.

The overall design of LAF is illustrated in Figure 1. The fundamental principle is that the user controls the representation format for linguistic annotations, which is mappable to the data model. This

mapping is accomplished via a rigid "dump" format, isomorphic to the data model and intended primarily for machine rather than human use.

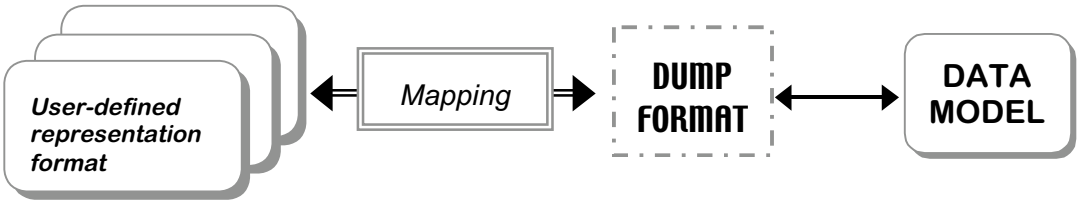

Figure 1. Overall LAF architecture

4 Dump format specification

The data model is built around a clear separation of the *structure* of annotations and their *content*, that is, the linguistic information the annotation provides. The model therefore combines a structural *meta-model,* that is, an abstract structure shared by all documents of a given type (e.g. syntactic annotation), and a set of *data categories* associated with the various components of the structural meta-model.

The structural component of the data model is a feature structure graph capable of referencing *n*-dimensional regions of primary data as well as other annotations. The choice of this model is indicated by its almost universal use in defining general-purpose annotation formats, including the Generic Modeling Tool (GMT) (Ide and Romary, 2001, 2002) and Annotation Graphs (Bird and Liberman, 2001). A small inventory of logical operations over annotation structures is specified, which define the model's abstract semantics. These operations allow for expressing the following relations among annotation fragments:

- *Parallelism:* two or more annotations refer to the same data object;
- *Alternatives:* two or more annotations comprise a set of mutually exclusive alternatives (e.g., two possible part-of-speech assignments, before disambiguation);
- *Aggregation:* two or more annotations comprise a list (ordered) or set (unordered) that should be taken as a unit.

The feature structure graph is a graph of elementary structural nodes to which one or more data category/value pairs are attached, providing the semantics of the annotation. LAF does not provide definitions for data categories. Rather, to ensure semantic coherence we specify a mechanism for the formal definition of categories and relations, and provide a *Data Category Registry* of predefined categories that can be used directly in annotations. Alternatively, users may define their own data categories or establish variants of categories in the registry; in such cases, the newly defined data categories will be formalized using the same format as definitions available in the registry.

5 Implementation

5.1 Dump format

The dump format is instantiated in XML. Structural nodes are represented as XML elements. The XML-based GMT will serve as a starting point for defining the dump format. Its applicability to diverse annotation types, including terminology, dictionaries and other lexical data (Ide, *et al.*, 2000), morphological annotation (Ide and Romary, 2002) and syntactic annotation (Ide and Romary, 2001b, 2003) demonstrates its generality.

As specified by the LAF architecture, the GMT implements a feature structure graph. Structural nodes in the graph are represented with the XML element `<struct>`. `<brack>` and `<alt>` elements

are provided as grouping tags to handle aggregation (grouping) and alternatives (disjunction), as described above. A <feat> element is used to express category/value pairs. All of these elements are recursively nestable. Therefore, hierarchical relations among annotations and annotation components can be expressed via XML syntax via element nesting. Other relations, including those among discontiguous elements, rely on XML's powerful inter- and intra-document pointing and linkage mechanisms. Because all annotations are stand-off (i.e., in documents separate from the primary data and other annotations), the same mechanisms are used to associate annotations with both "raw" and XML-tagged primary data and with other annotations.

The final XML implementation of the dump format may differ slightly from the GMT, in particular where processing concerns (e.g. ease of processing elements vs. attributes vs. content) and conciseness are applied. However, in its general form the above are sufficient to express the information required in LAF. For examples of morphological and syntactic annotation in GMT format, see Ide and Romary, 2001a; 2003; and Ide and Romary, 2001b.

5.2 Data Categories

To make them maximally interoperable and consistent with existing standards, RDF schemas can be used to formalize the properties and relations associated with data categories. Instances of the categories themselves will be represented in RDF. The RDF schema ensures that each instantiation of the described objects is recognized as a sub-class of more general classes and inherits the appropriate properties. Annotations will reference the data categories via a URL identifying their instantiations in the Data Category Registry itself. The class and sub-class mechanisms provided in RDFS and its extensions in OWL will also enable creation of an ontology of annotation classes and types.

For example, the syntactic feature defined in the ISLE/MILE format for lexical entries (Calzolari, *et al.* 2003) can be represented in RDF as follows[1]:

```
<rdf:RDF>
<Phrase rdf:ID="Vauxhave">
   <hasSynFeature>
     <SynFeature>
        <hasSynFeatureName rdf:value="aux"/>
        <hasSynFeatureValue rdf:value="have"/>
      </SynFeature>
   </hasSynFeature></Phrase>
</rdf:RDF>
```

Once declared in the Data Category registry, annotations or lexicons can reference this object directly, for example:

```
<Self rdf:ID="eat1Self">
  <headedBy
   rdf:resource="http://www.DCR /Vauxhave"/>
</Self>
```

For a full example of the use of RDF-instantiated data categories, see Ide, *et al.,* in this volume.

Note that RDF descriptions function much like class definitions in an object-oriented programming language: they provide, effectively, templates that describe how objects may be instantiated, but do not constitute the objects themselves. Thus, in a document containing an actual annotation, several objects with the same type may be instantiated, each with a different value. The RDF schema ensures that each instantiation is recognized as a sub-class of more general classes and inherits the appropriate properties.

A formally defined set of categories will have several functions: (1) it will provide a precise semantics for annotation categories that can be either used "off the shelf" by annotators or modified to serve specific needs; (2) it will provide a set of reference categories onto which scheme-specific names can be mapped; and (3) it will provide a point of departure for definition of variant or more precise categories. Thus the overall goal of the Data Category Registry is not to impose a specific set of categories, but rather to ensure that the semantics of data categories included in annotations (whether they exist in the Registry or not) are well-defined and understood.

[1] For brevity, this representation does not include the full information necessary for the RDF representation.

6 Conclusion

In this paper we describe the Linguistic Annotation Framework under development by ISO TC37/SC 4 WG1-1. Its design is intended to allow for, on the one hand, maximum flexibility for annotators, and. on the other, processing efficiency and reusability. This is accomplished by separating user annotation formats from the exchange/processing format. This separation ensures that pre-existing annotations are compatible with LAF, and that users have the freedom to design specific schemes to meet their needs, while still conforming to LAF requirements.

LAF provides for the use of any annotation format consistent with the feature structure-based data model that will be used to define the pivot format. This suggests a future scenario in which annotators may create and edit annotations in a proprietary format, transduce the annotations using available tools to the pivot format for interchange and/or processing, and if desired, transduce the pivot form of the annotations (and/or additional annotation introduced by processing) back into the proprietary format. We anticipate the future development of annotation tools that provide a user-oriented interface for specifying annotation information, and which then generate annotations in the pivot format directly. Thus the pivot format is intended to function in the same way as, for example, Java byte code functions for programmers, as a universal "machine language" that is interpreted by processing software into an internal representation suited to its particular requirements. As with Java byte code, users need never see or manipulate the pivot format; it is solely for machine consumption.

Part of the work of SC4 WG1-1 is to provide development resources, including schemas, design patterns, and stylesheets, which will enable annotators and software developers to immediately adapt to LAF. Example mappings, e.g., for XCES-encoded annotations, will also be provided. In this way, we hope to realize the goal of harmonized and reusable resources in the near future.

References

Bird, S. and Liberman, M. (2001). A formal framework for linguistic annotation. *Speech Communication,* 33:1-2, 23-60.

Bunt. H. and Romary, L. (2002). Towards Multimodal Content Representation. *Proceedings of the Workshop on International Standards for Terminology and Language Resource Management*, Las Palmas.

Calzolari, N., Bertagna, F., Lenci, A., Monachini, M., 2003. *Standards and best Practice for Multilingual Computational Lexicons and MILE (Multilingual ISLE Lexical Entry)*, ISLE Computational Lexicon Working Group deliverables D2.2 – D3.2, Pisa.

Ide, N. and Romary, L. (2001a). Standards for Language Resources, *IRCS Workshop on Linguistic Databases*, Philadelphia, 141-49.

Ide, N. and Romary, L. (2001b). A Common Framework for Syntactic Annotation. *Proceedings of ACL'2001,* Toulouse, 298-305.

Ide, N. and Romary, L. (2002). Standards for Language Resources. *Proceedings of the Third Language Resources and Evaluation Conference* (LREC), Las Palmas, Canary Islands, Spain, 839-44.

Ide, N. and Romary, L. (2003). Encoding Syntactic Annotation. In Abeille, A. (ed.). *Treebanks: Building and Using Syntactically Annotated Corpora*. Dordrecht: Kluwer Academic Publishers (in press).

Ide, N., Kilgarriff, A., and Romary, L. (2000). A Formal Model of Dictionary Structure and Content. *Proceedings of Euralex 2000,* Stuttgart, 113-126.

Ide, N., Lenci, A., And Calzolari, N. (2003). RDF Instantiation of ISLE/MILE Lexical Entries. This volume.

Ide, N., Romary, L, and De la Clergerie, E. (2003). International Standard for a Linguistic Annotation Framework. *Proceedings of NAACL'03 Workshop on Software Engineering and Architecture of Language Technology Systems* (to appear).