Chapter 9

TOWARDS INTERNATIONAL STANDARDS FOR LANGUAGE RESOURCES

Nancy Ide
Vassar College, USA
ide@cs.vassar.edu

Laurent Romary
LORIA-INRIA, France
Laurent.Romary@loria.fr

Abstract This chapter describes the Linguistic Annotation Framework (LAF) developed by the International Standards Organization TC37 SC4, which is to serve as a basis for harmonizing existing language resources, as well as developing new ones. We then describe the use of the LAF to represent the American National Corpus and its linguistic annotations.

Keywords Language resources; Standards; Corpora; Linguistic annotation; ISO; American National Corpus.

1 Introduction

As noted in Cole et al. (1997), years of research and development in computational linguistics and language engineering have yielded many stable results, which have in turn been integrated into language processing applications and industrial software. Especially over the past 15 years, researchers and developers have increasingly understood the need to define common practices and formats for linguistic resources, which serve Human Language Technologies (HLT) development as the primary source for statistical language modelling. To answer this need, numerous projects have been launched to lay the basis for standardization of resource representation

L. Dybkjær et al. (eds.), Evaluation of Text and Speech Systems, 263–284.

and annotation – e.g., the Text Encoding Initiative (TEI) (http://www.tei-c.org), the Corpus Encoding Standard (CES and XCES) (http://www.xml-ces.org), the Expert Advisory Group on Language Engineering Standards (EAGLES) and the International Standard for Language Engineering (ISLE) (http://www.ilc.cnr.it/EAGLES96/home.html), as well as software platforms for resource creation, annotation, and use – MULTEXT (http://www.lpl.univ-aix.fr/projects/multext), LT XML (http://www.ltg.ed.ac.uk/software/xml), GATE (http://gate.ac.uk/), NITE (http://www.dfki.de/nite/main.html), ATLAS (http://www.nist.gov/speech/atlas/). However, although in practice consensus has begun to emerge, definitive standards have not yet been put in place. In large part this is as it should be: advances in technology together with the emergence of a solid body of web-based standards have dramatically impacted and redefined many of our ideas about the ways in which resources will be stored and accessed over the past several years. Perhaps more importantly, the ways in which language data – together with "communicative" data of any kind, including gesture, facial expression, and speech – are processed and analyzed will certainly continue to change, as more and more emphasis is put on immediate processing of (often multimodal) streamed data. Whatever the scenario, though, if we intend to make HLT work in the larger arena of universal availability and accessibility, data, its annotations, and processing results will have to be represented in some way that allows exploitation by the full array of language processing technologies.

It has been argued that attempting standardization for language resources and surrounding information is premature, and the evolving nature of the domain and technology certainly speaks to that claim. But the growth of the web and the explosion in the number of electronic documents to be handled and maintained within the industrial sector has created an immediate and urgent need for generic language processing components for document indexing and classifying, information extraction, summarization, topic detection, etc., in both mono- and multilingual environments, together with robust machine translation and facilities for man-machine multimodal communication. While progress will continue, the field has nonetheless reached a point where we can see clear to a reasonable representation and processing model that should fulfil the needs of HLT for at least the foreseeable future. Indeed, commonality that can enable flexible use and reuse of communicative data is essential for the next generation of language processing applications, if we are to build a global information environment. It is therefore critical at this time to move toward standardization, and in particular, to do this in an internationally accepted framework.

It is in this context that a committee of the International Standards Organization (ISO), TC 37/SC 4, has been established to develop standards for *language resources management*, with the aim of building on existing technologies and schemes to codify best practices as a set of standards for representing and processing language-related information, as a means to leverage the growth of language engineering. Fully aware that its activities will be necessarily on going and evolving, the committee has set out the following general goals:

- To provide means to use and reuse linguistic data across applications, at all levels of linguistic description from surface mark-up of primary sources to multilayered processing results
- To facilitate maintenance of a coherent document life cycle through various processing stages, so as to enable enrichment of existing data with new information and the incremental construction of processing systems

2 Background

Before initiating any standardizing activity, it is necessary to identify its scope and relation to past and/or ongoing activities. As a starting point, Figure 1 describes the general "ecology" of language resources and the interdependencies required for their management.

Primary resources may be texts, spoken data, multimodal data (e.g., hand motion, eye gaze, perceptual settings, etc.). *Linguistic information* consists of annotations (ranging from phonetic and morpho-syntactic annotation to discourse level annotations, such as reference chains, dialogue structure, etc.) associated with a segment or segments of a primary resource or other descriptive layer.[1] *Lexical and knowledge structures* may be linked to primary resources and annotations, or created from primary resources; they are most often used to support linguistic analysis, including annotation. As such, they often are the source of information that is used for linguistic annotation. *Metadata* can be regarded as another type of annotation associated with a document containing primary or annotation data, which identifies and describes the resource. Finally, *links* and *access protocols* provide the mechanisms for representing and accessing language resources.

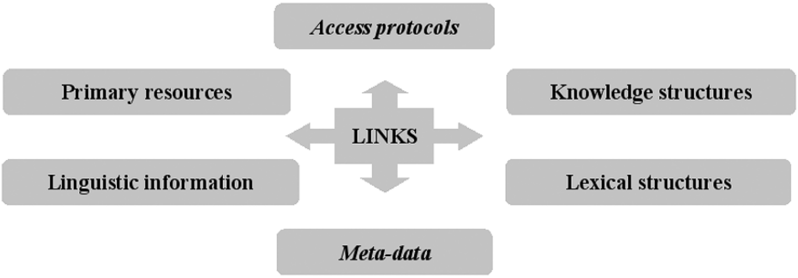


Figure 1. Ecology of language resources.

Over the last 20 years, numerous projects and initiatives have worked towards the development of standards for one or more of the components pictured above, as well as for a general architecture that would enable efficient representation of the resources themselves together with the "links" establishing the interdependencies among them. Among the most notable are the TEI, CES, and XCES, and MATE/NITE for the representation of primary data and annotations; EAGLES/ISLE for annotation content; OLIF (http://www.olif.net/), SALT (http://www.loria.fr/projets/SALT/), and ISLE for various kinds of lexical/terminological data; RDF/OWL and Topic Maps for knowledge structures; Dublin Core and the Open Archives Initiative (OAI) (http://www.openarchives.org/) for general metadata; MPEG7, IMDI, and OLAC for domain-specific metadata; Corba (http://www.corba.org/) and the W3C's SOAP (http://www.w3.org/TR/soap/) and web services work for access protocols; and MULTEXT, Edinburgh's LT framework, TIPSTER (http://www.fas.org/irp/program/process/tipster.htm), GATE, and ATLAS for general architecture. Most of these projects actually address several of what we can regard as the multiple "dimensions" of language resource representation, including (at least) the following:

Rendering formats and mechanisms, such as SGML, XML, Lisp-like structures, annotation graphs, or a particular database format.

Annotation content, including categories of annotation information for linguistic phenomena (e.g., modality, aspect, etc.) and the values that can be associated with each category.

General architectural principles for language resources, such as the now widely- accepted notions of pipeline architecture and stand-off annotation.

Even here, there are interdependencies: for example, the choice of a representation format will have repercussions for content, first of all because relations among pieces of information may be expressed implicitly through the structures provided by the format, the most common of which is a hierarchical structure for grouping and/or defining part/whole relations. Some formats impose other constraints – for example, Lisp-like formats provide a hierarchical structure but do not readily accommodate labelling the structures to distinguish their function (e.g., grouping, listing alternatives, etc.), as one might do in XML by simply giving a tag a meaningful name. Similarly, implementing stand-off annotation with XML dictates use of XML paths, pointers, and links. As a result, format and content have in past projects often been treated as a whole, rather than addressing them separately.

Despite the numerous projects and initiatives that have sought to establish standards for various aspects of linguistic annotation, there remains no universally accepted set of practices and categories, and there continues to be considerable reinvention of the wheel within the international community. This begs the question: why should the ISO effort succeed where others have failed? There are several answers to this question, the most notable of which is the evolution of technology, both in terms of the availability of accepted frameworks that operate within the web context, including primarily World Wide Web Consortium (W3C) standards, such as XML and RDF/OWL, together with cross-platform/web-adaptable software development tools, such as Java. However, the technological advances resulting from development of the web has done more than provide us with widely accepted standards for language data representation. The shift from stand-alone applications to an environment where both data and software is distributed over the web has dramatically impacted the ways in which we create and represent language resources and their annotations, as well as the kinds of information we want to represent. The enhanced potential to exploit the web to share, merge, and compare language data has itself encouraged widespread adoption of W3C representation standards, and indeed, the web itself has come to be regarded as a virtually infinite "corpus" of multilingual and multimodal data. In addition, in the context of the web certain language processing applications – e.g., information retrieval and extraction, summarization, etc., together with applications that handle multimodal data – have taken the foreground, and the kinds of information that we are most interested in identifying and processing have evolved in tandem. The web has also spawned heightened interest in what we can regard as "on the fly" annotation and analysis for streamed data, and more generally, a need to support incremental annotation at various linguistic levels.

Attempts to standardize linguistic content categories and their values have always been plagued by the thorny problem of varying linguistic theories and application needs: some *de facto* standards, such as WordNet for semantic annotation, have emerged, but there is relatively little commonality in this area beyond these few exceptions despite massive efforts, such as the EAGLES/ISLE project. The forces driving new interest in harmonization of annotation content are similar to those driving standardization for data representation: the existence of the web and the promise of a "semantic web" demand common terminology for every level of description, as the recent efforts to develop standard metadata categories and ontologies demonstrate. The ontology efforts also show how difficult content standardization is to achieve. So, while we have increased motivation to develop linguistic content categories, and possibly a better base than at any time in the past from which to proceed, this aspect of language resource standardization can only be approached cautiously and, likely, far more slowly than resource representation.

With a sounder technological base and a clearer idea of where we need to go, yet another standardization effort seems to be in order. It is important to note, however, that the ISO effort builds to the extent possible on previous efforts, adopting the parts it can and extending or modifying them as seems necessary, and taking advantage of the incremental convergence of opinion on various aspects of the process that has directly resulted from attempts at standardization and/or commonality in the past. To this end, the ISO group has established collaborations with major standardizing groups, including most of the prior initiatives enumerated above, as well as others involved in standardization activities, in order to ensure that the development of ISO standards for language resource management both incorporates and reflects existing practice and informs ongoing work within these other groups. In addition, the ongoing work within the ISO committee is continually presented at major conferences and workshops so that the community is aware of our work and can comment and contribute to the effort.

The "incremental view" of standardization, wherein standards are developed over a series of iterations that potentially span decades, informs both the work within ISO/TC 37/SC 4 and the place of its work in the overall scheme. The standards developed by this ISO sub-committee may not be the final word on language resource representation and management, but they will, we hope, take a necessary step toward that goal. Our work, like the creation of the web-based infrastructure being developed by W3C and others, is best seen as part of a development process that can be compared to building a brick wall: we add brick by brick, layer by layer, and occasionally develop some infrastructural component that adds a significant piece to the overall construction. We are not sure when or where this process will end, but each effort is required for eventual completion.

3 The Linguistic Annotation Framework

The Linguistic Annotation Framework (LAF) is intended to provide a standard infrastructure for representing language resources and their annotations that can serve as a basis for harmonizing existing resources, as well as developing new ones.

Annotation of linguistic data may involve multiple annotation steps, for example, morpho-syntactic tagging, syntactic analysis, entity and event recognition, semantic annotation, coreference resolution, discourse structure analysis, etc. Annotation at lower linguistic levels typically serves as input to the higher-level annotation process in an incremental process. Depending on the application intended to use the annotations, lower-level annotations may or may not be preserved in a persistent format. For example, information extraction software often annotates linguistic features required to generate the final annotation, without preserving the intermediate information. In other situations, the annotation process may not be strictly incremental. For example, when handling streamed data (text, video, and audio, a stream of sensor readings, satellite images, etc.) the processor analyzes language data in a linear, time-bound sequence, and therefore annotations may be temporarily partial during processing if long-distance dependencies between seen and unseen segments of the data exist.

At present, most annotated resources are static entities used primarily for training annotation software, as well as corpus linguistics and lexicography. However, in the context of the Semantic Web, annotations for a variety of higher-level linguistic and communicative features will increasingly be preserved in web-accessible form and used by software agents and other analytic software for inferencing and retrieval. This dictates that the LAF not only relies on web technologies (e.g., RDF/OWL) for representing annotations, but also that "layers" of annotations for the full range of annotation types (including named entities, time, space, and event annotation, annotation for gesture, facial expression, etc.) are at the same time separable (so that agents and other analytic software can access only those annotation types that are required for the purpose, and mergeable (so that two or more annotation types can be combined where necessary). They may also need to be dynamic, in the sense that new and/or modified information can be added as necessary.

The LAF consists of two major components:

1. An abstract data model and a concrete representation format isomorphic to the model
2. A mechanism for defining and using linguistic categories and values

Each of these components is covered in the following sections.

3.1 Architecture and Abstract Model

In order to ensure that the LAF architecture reflects state-of-the-art methods drawn from consensus of the research community, a group of experts[2] was convened in November, 2002, to lay out its overall structure. The group, which included researchers with extensive experience in the development of annotation schemes at a variety of linguistic levels together with developers of major resource-handling software (GATE, ATLAS, Edinburgh LT tools), defined the general architecture pictured in Figure 2.

The fundamental principle underlying the LAF architecture is that the user controls the representation format for linguistic resources and annotations, using any desired scheme (XML, LISP structures, or any other format). The only restriction applied to the user format is that it must be mappable to an *abstract 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.

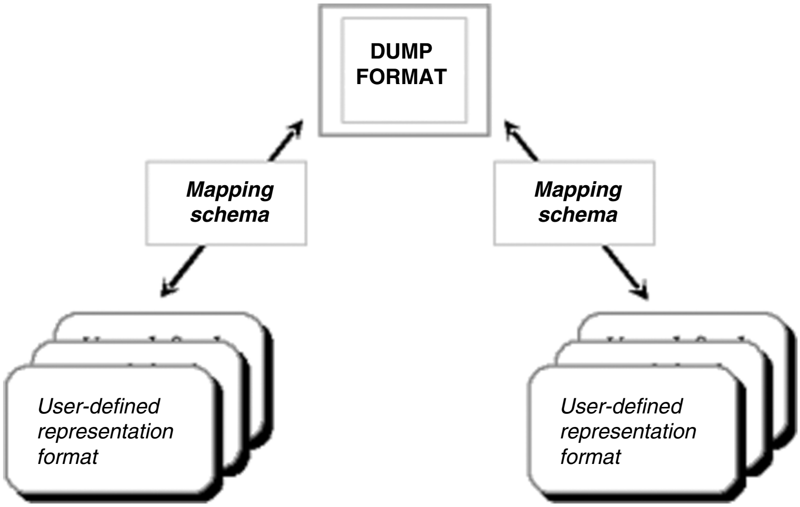

DUMP FORMAT
Mapping schema
Mapping schema
User-defined representation format
User-defined representation format

Figure 2. LAF architecture.

To guide the LAF development, the following general principles were outlined by the group of experts:

- The data model and document form are distinct but mappable to one another.
- The data model is parsimonious, general, and formally precise.
- The document form is largely under user control.
- The mapping between the flexible document form and data model is via a rigid dump-format. The responsibility of converting to the dump format is on the producer of the resource.
- Mapping is operationalized via either a schema-based data-binding process or schema-derived stylesheet mapping between the user document and the dump format instantiation. The mapping from document form to the dump format is documented in an XML Schema (or the functional equivalent thereof) associated with the dump format instantiation.
- It must be possible to isolate specific layers of annotation from other annotation layers or the primary (base) data; i.e., it must be possible to create a dump format instantiation using stand-off annotation.
- The dump format supports stream marshalling and unmarshalling.

The dump format is isomorphic to the underlying abstract data model, which is built upon a clear separation of the *structure* of linguistic information (including annotations and knowledge structures) and *content*, i.e., the linguistic information itself. A document and its annotations form a directed graph capable of 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, as specified in ISO 10646/Unicode. 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 labelled with feature structures containing the annotation content relevant to the data identified by the edge. 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; Ide and Romary, 2002; Ide and Romary, 2004b) and Annotation Graphs (Bird and Liberman, 2001). All annotations are stand-off – i.e., represented in documents separate from the primary data and other annotations – in order to support incremental annotation and separability of different annotation levels.

The graph of feature structures contains elementary structural nodes to which one or more feature structures are attached, providing the semantics ("content") of the annotation. A small inventory of logical operations (e.g., disjunction, sets) over the feature structures is specified, which define the model's abstract semantics. These operations provide the same expressive power as those defined for general-purpose, typed feature structures. Semantic coherence is provided by a registry of features maintained in RDF/OWL format, as described in Section 3.2. Users may define their own data categories or establish variants of categories in the registry. In the latter case, the newly defined data categories are formalized using the same format as definitions available in the registry. A schema providing the mapping of categories used in the document to categories in the registry and the formal specification of newly-defined categories is associated with the dump format instantiation.

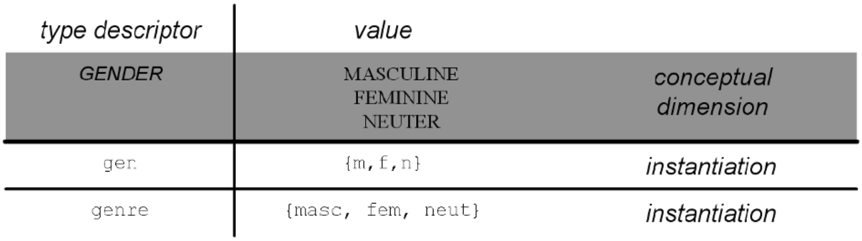
In the LAF scenario, the dump format is invisible to users; users work only with their own formats, and transduce to and from the dump format only for processing and exchange. Thus, each site need only define a mapping between an in-house format and the dump format in order to use resources produced by any other site.

3.2 Data Category Registry

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[3], 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, 2004a).

Differences in approach to language resources and among individual system objectives inevitably lead to variations in data category definitions and data category names. The use of uniform data category names and definitions within the same resource domain (e.g., among terminological, lexicographical, and text corpus resources), at least at the interchange level, contributes to system coherence and enhances the reusability of data. Procedures for defining data categories in a given resource domain should also be uniform in order to ensure interoperability.

We define a *data category* as an elementary descriptor used in a linguistic annotation scheme. In feature structure terminology, data categories include both attributes (hereafter called *type descriptors*), such as SYNTACTIC CATEGORY and GRAMMATICAL GENDER, as well as a set of associated atomic *values* taken by such attributes, such as NOUN and FEMININE. In both cases we distinguish between the abstraction (concept) behind an attribute or value, and its realization as some string of characters or other object. Figure 3 provides an overview of these relationships. Whereas there is only one concept for a given attribute or value, there may be multiple instantiations.



type descriptor	*value*	
GENDER	MASCULINE FEMININE NEUTER	*conceptual dimension*
ger	{m,f,n}	*instantiation*
genre	{masc, fem, neut}	*instantiation*

Figure 3. Data category overview.

The DCR under development within ISO/TC 37/SC 4 is built around this fundamental concept/instance distinction. In principle, the DCR provides a set of reference concepts, while the annotator provides a *Data Category Specification* (DCS) that comprises a mapping between his or her scheme-specific instantiations and the concepts in the DCR. As such, the DCS provides documentation for the linguistic annotation scheme in question. The DCS for a given annotation document/s is included or referenced in any data exchange to provide the receiver with the information required to interpret the annotation content or to map it to another instantiation. Semantic integrity is guaranteed by mutual reference to DCR concepts.

To serve the needs of the widest possible user community, the DCR must be developed with an eye toward multilingualism. The DCR will support multiple languages by providing the following:

- Reference definitions for data categories in various languages
- Data element names for the data categories in various languages
- Description of usage in language-specific contexts, including definitions, usage notes, examples, and/or lists of values (e.g., GENDER takes the values *masculine, feminine* in French; *masculine, feminine, neuter* in German)

In addition, to both accommodate archival data and ensure semantic integrity, a mapping of data categories instantiated in the DCR to categories and values in well-known projects and initiatives will be provided.

The creation of a single global DCR for all types of language resources treated within TC 37 provides a unified view over the various applications of the resource. However, for the purposes of both category creation and DCR access, the DCR will be organized according to *thematic views*, i.e., domains of activity, which include specialized subsets of the information in the registry. Given the ongoing activities within TC 37, we can envisage definable subsets of the DCR for at least the following: terminological data collection, various types of linguistic annotation (morpho-syntactic, syntactic, discourse level, etc.), lexical representation for both NLP-oriented and traditional lexicography, language resource metadata, and language codes.

Figure 4 illustrates the relationship between data category specifications and the DCR. The patterned cells correspond to individual DCSs. Some data categories are relevant to a single domain, while others are common to multiple domains: for example, *sense number* is probably specific to lexicographical resources, but linguistic categories, such as *part of speech*, *grammatical gender, grammatical number*, etc. have wider application. Each thematic domain contributes all its data categories the global DCR, while at the same time identifying those data categories that it shares with other domains.

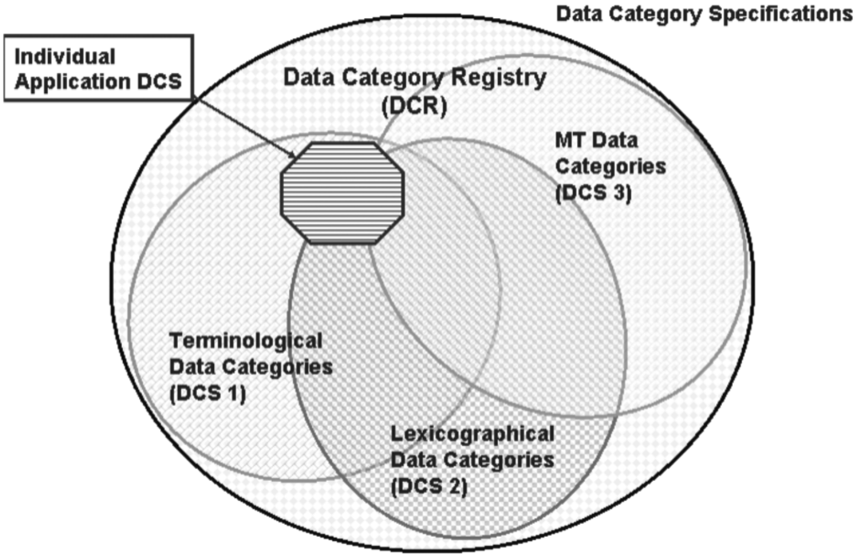

Data Category Specifications
Individual Application DCS
Data Category Registry (DCR)
MT Data Categories (DCS 3)
Terminological Data Categories (DCS 1)
Lexicographical Data Categories (DCS 2)

Figure 4. Relation of data category selections to the DCR.

The oval shapes in the Venn diagram represent DCS subsets. A smaller subset can be selected from the domain DCS for use in a given application, as represented by the octagon in Figure 4. Note that while some of the data categories contained in this subset are common to several different domains, this application is wholly contained within the DCS for terminological entries, so we can conclude that it is designed for use with a terminological application.

We intend to proceed cautiously, implementing categories that are widely used and relatively low-level, to ensure acceptance by the community. By building up slowly, the DCR should eventually contain a wide range of data categories, with their complete history, data category description, and attendant metadata. It would then be possible to specify a DCS (see previous section) for different thematic domains and an ontology of relations among them. In the short term, it is likely unreasonable to define such an ontology until there is greater awareness and consensus at the international level. No choice should be made in the definition of the DCR that would hamper further work in this direction.

So far, we have defined a preliminary template for data category definitions to be used as an underlying model for the DCR (ISO DIS 12620 under ISO committee TC 37/SC 3), which can also serve as a model for manipulation and transmission of proprietary data categories within the language engineering community. The heart of a data category description is the *Conceptual Entry* section, which we define to include the following fields:

ENTRY IDENTIFIER used for interchange of data category;
DEFINITION reference definition for the category, language, and theory neutral to the extent possible;
EXPLANATION additional information about the data category not relevant in a definition (e.g., more precise linguistic background for the use of the data category);
EXAMPLE illustration of use of the category, excluding language specific usages (documented elsewhere);
SOURCE may refine definition, explanation, or example to indicate the source from which the corresponding text has been borrowed or adapted;
STATUS may refine definition to indicate approval, acceptability, or applicability in a given context;
PROFILE relates the current data category to one or several views (e.g., Morphosyntax, Syntax, Metadata, Language description, etc.);
CONCEPTUAL RANGE relates the category to the set of possible values (expressed as a list of data categories). A datatype may be provided instead of a list of values;
NOTE additional information excluding technical information that would normally be described within explanation;
BROADER CONCEPT generic pointer to a more general data category (e.g., from common noun to noun).

3.3 Using the DCR

The purpose of the DCR is to promote greater usability and reusability of annotated language resources and increased semantic integrity for information in annotation documents by providing a set of formally defined reference categories. “Formal definition” in this context includes natural language definitions for each category accompanied by specification of the possible values each category may take. At present, we envision instantiation of the DCR as a simple database in which each entry is either a type descriptor or value. Data categories will be referenced either by the DCR entry identifier, or, since the DCR will be publicly available online, via a URI.

Note that this simple instantiation of the DCR makes no distinction in terms of representation between type descriptors and values; each is considered as a data category and provided with an entry identifier for reference. Only minimal constraints on their use in an annotation are specified – i.e., constraints on descriptor/value combinations given in the descriptor entry. The broader structural integrity of an annotation is provided by placing constraints on nodes in the annotation graph (as defined in the LAF architecture) with which a given category can be associated. For example, the structural graph for a syntactic constituency analysis would consist of a hierarchy of typed nodes corresponding to the non-terminals in the grammar, with constraints on their embedding, and with which only appropriate descriptor/value pairs may be associated. Node types (e.g., NP, VP), as well as associated grammatical information (e.g., tense, number) may all be specified with data categories drawn from the DCR.

A more formal specification of data categories can be provided using mechanisms, such as RDF Schema (RDFS) and the Ontology Web Language (OWL) to formalize the properties and relations associated with data categories. For example, consider the following RDF Schema fragment:

```
<rdfs:Class rdf:about="#Noun">
  <rdfs:label>Noun</rdfs:label>
  <rdfs:comment>Class for
          nouns</rdfs:comment>
</rdfs:Class>
<rdfs:Property rdf:about="#number">
   <rdfs:domain
      rdfs:resource="Noun"/>
   <rdfs:range
      rdf:resource="rdfs:#Literal"/>
</rdfs:Property>
```

This fragment defines a class of objects called “Noun” that has the property “number”. Note that the schema defines the classes but does not instantiate objects belonging to the class; instantiation may be accomplished directly in the annotation file, as follows (for brevity, the following examples assume appropriate namespace declarations specifying the URIs of schema and instance declarations)

```
<Noun rdf:about="Mydoc#W1">
     <number rdf:value="Plural"/>
</Noun>
```

where “Mydoc#W1” is the URI of the word being annotated as a noun. Alternatively, the DCR could contain instantiations of basic data elements, specifying values for properties, which can be referenced directly in the annotation. For example, the DCR could include the following instantiation:

```
<Noun rdf:ID="NMP">
    <number rdf:value="plural"/>
</Noun>
```

An annotation document could then reference the pre-defined instance as follows:[4]

```
<rdf:Description rdf:about="myDoc#W1">
    <POS rdf:resource="categories#NMS"/>
</rdf:Description>
```

An RDFS/OWL specification of data categories would enable greater control over descriptor/value use and also allow for the possibility of inferencing over annotations. RDFS/OWL descriptions function much like class definitions in an object-oriented programming language: they provide, in effect, templates that describe the properties of an object, specify constraints on which objects can provide the value for a given property, and specify super- and subclass relations among objects. For example, a general *dependent* relation may be defined for a verb object, which must have one of the possible values *argument* or *modifier*; *argument* can in turn have the possible values *subject*, *object*, or *complement*, etc., cf. the hierarchy in Figure 3, (Carroll et al., 2004). In a document containing a syntactic annotation, several objects with the type *argument* may be instantiated, each with a different value. Based on the RDFS/OWL definition, each instantiation of *argument* is recognized as a subclass of *dependent* and inherits the appropriate properties.

Definition of a precise hierarchy of linguistic categories and properties is a massive undertaking, and it is far from obvious that such a hierarchy could be agreed upon within the community. Therefore, we are proceeding cautiously to define hierarchical relations among categories, and leaving the bulk of this activity to users of the DCR. We will provide a library of RDF/OWL specifications describing hierarchical relations together with value constraints, interdependencies, etc., that can be used as desired by annotators. We expect that the library will be built up gradually from our initial descriptions and the contributions of users.

It cannot be overemphasized that the goal of the DCR is not to impose a specific set of categories, but rather to ensure that the semantics of data categories included in annotations are well-defined, either by referring to categories that are formally described in the DCR or by formal definition of new or variant categories. The DCR, at least at the outset, can only help us to move toward commonality in annotation content, which is becoming more and more essential as annotated language data is increasingly distributed over multiple sites and accessible via the web.

In the end, the DCR will come into widespread use only if it is easy for annotators to use and provides useful categories for various kinds of resource annotation. Ease of use can be assured by providing ready-to-use templates for reference to the DCR from within annotation documents, enabling immediate web access to definitions in a clear and concise format, and, perhaps above all, ensuring that at least a few highly visible projects use DCR references. The initial inclusion of categories that are for the most part relatively atomic and universally accepted is a move towards ensuring their usefulness for linguistic annotation, but, if the DCR is to be truly successful, it will also be necessary to include and demonstrate the use of categories that have become, for better or worse, *de facto* standards defined by widely used resources. The obvious example is WordNet: whatever its shortcomings for NLP, WordNet is the most universally used resource in the field, and there are now over 30 WordNets in different languages built around the same categories and concepts. One way to bring the DCR into general use is to implement a "DCR-aware" version of WordNet that specifies a mapping of WordNet categories to the DCR, and, on the other hand, ensure that WordNet-specific categories (e.g., synset) and all categories used in WordNet (e.g., meronym, hypernym, etc.) are in fact included in the DCR. Similarly, a mapping of categories in FrameNet, which is now also being replicated for other languages, and other existing or developing "standards", such as the EAGLES morpho-syntactic categories, TIME-ML (http://www.timeml.org), etc., can be made available via the DCR website. In this way, annotators will become aware of DCR categories and have real examples demonstrating DCR use.

4 Putting it All Together

To illustrate how the LAF principles are applied in practice, consider an interchange scenario between two users ("A" and "B"), each having his/her own annotation scheme for a given annotation layer, and a third user ("C") who wants to use both A's and B's annotations. Such a scenario is in fact typical within evaluation campaigns, such as PARSEVAL.

A and B apply LAF by mapping the descriptors used in their respective annotation schemes to categories in the DCR. The mapping is specified using an RDF/OWL schema, for which a template or automatic generation tool is available on the DCR website. If categories used in the user-specific annotation scheme are not included in the DCR, or if a DCR definition for a given category requires modification or extension, the new or variant categories are fully defined in the schema (again using a template or tool available on the DCR website).

Next, the user format is transduced to a LAF representation. The transduction may reveal that some of the annotation information in the user's scheme is implied by its structure; for example, in the Penn Treebank (PTB) syntactic annotation, the "subject" relation between a noun phrase and a verb phrase is implied by their relative positions in the parse tree represented by the LISP format, while the "object" relation is given explicitly (via an NP-Obj label) because the position of an NP in the tree is less definitively indicative of its semantic role. Similarly, embedded "S-units" in the PTB imply what is often called an "xcomp" relation, which in turn (implicitly, in the PTB) inherits its subject from the S-unit within which it is nested. In order to use such implicit information, the software must be aware that, for instance, the first NP within an S is to be considered the subject. However, it should not be expected that user C's software is designed to make this inference, and therefore LAF compliance requires that such information be made explicit by the creator of the original scheme when transducing to LAF format.[5]

The transduction process demands familiarity with the LAF XML format and moderate computational expertise to create a transduction script. LAF-compliaint annotations are represented in a generic XML format for specifying edges (using a <struct> element, historically so-named to stand for "structural node") and the associated feature structures; as such, the XML elements provide the structure of the annotation but do not include any information concerning annotation content. The actual content of the annotation is provided in the attribute/value pairs within the feature structure.[6]

The transduction process therefore involves user-specific structures (e.g., nested parentheses in the PTB LISP example) to XML <struct> elements, and filling attribute value slots in the feature structure encoding with the appropriate labels. Because all LAF annotation documents are stand-off, it also may involve disentangling data and annotations, and providing XPointer links from edges (<struct> elements) in the annotation document to the primary data.

An example of a PTB transduction to LAF format is given in Figures 5 and 6. Each <struct> element corresponds to an edge in the graph, traversing the indicated span in the primary data. <feat> elements provide the feature/value pairs associated with the immediate parent node.[7] Note that in this example, XML embedding of <struct> elements reflects the constituency relations among the edges, reflecting the LISP tree-structure. We take advantage of the fact that XML processors will reconstruct the implied tree structure from the embedding, while at the same time we are providing sufficient information to reconstruct it automatically from the values given in the TARGET attributes if XML processing is unavailable or inapplicable.

When user C obtains the LAF version of A's and B's annotations, the only processing requirement is that his tool understands the dump format to extract the annotation information in each one, either in order to use them directly in an application or transduce them to an in-house format of his own. Because

```
((S (NP-SBJ-1 Paul)
    (VP intends)
    (S   (NP-SBJ *-1)
       (VP  to
          (VP  leave
             (NP IBM))))
.))
```

Figure 5. PIB annotation of "Paul intends to leave IBM".

```
<struct target="xptr(substring(/p/s[1]/text(),1,26))">
 <feat type="syntacticCategory">S</feat>
 <struct id="s0" target="xptr(substring(/p/s[1]/text(),1,4))">
   <feat type="syntacticCategory">NP</feat>
   <feat type="syntacticFunction">subject</feat>
 </struct>
 <struct target="xptr(substring(/p/s[1]/text(),5,7))">
   <feat type="syntacticCategory">VP</feat>
 </struct>
 <struct target="xptr(substring(/p/s[1]/text(),12,12))">
   <struct target="s0"/>
   <struct>
    <feat type="syntacticCategory">VP</feat>
    <struct target="xptr(substring(/p/s[1]/text(),15,9))">
      <feat type="syntacticCategory">VP</feat>
      <struct target="xptr(substring(/p/s[1]/text(),21,3))">
        <feat type="syntacticCategory">NP</feat>
      </struct>
    </struct>
   </struct>
 </struct>
</struct>
```

Figure 6. Dump format instantiation of “Paul intends to leave IBM”.

both user A and B have provided a mapping of their respective categories in the RDF/OWL schema that accompanies the LAF-compliant annotation documents, user C can readily translate scheme-specific categories, such as “NP” to his own category designation, if they differ. So, for example, if user A uses “NP” for noun phrases, and user B uses “Nominal”, then if both A’s and B’s RDF/OWL schemas map these two designations to a common DCR category, user C knows that the two notations represent the same concept. User C, in turn, can map A’s and B’s notations to his own notation for that concept, if desired.

4.1 A Case Study: The ANC

The American National Corpus (ANC) project (http://AmericanNationalCorpus.org), which is creating a 100 million word corpus of American English comparable to the British National Corpus, is representing its data and annotations in accordance with the LAF specifications. The ANC is being heavily annotated for a variety of linguistic information, including morpho-syntax, syntax, named entities, semantics (WordNet sense tags and FrameNet frames), etc., and the project is providing multiple alternative annotations at each level produced by different automatic annotation tools. In order to accommodate the layering of several different POS taggings, noun, and verb chunks, dependency and constituency parse annotation schemes, and named entity annotations, and in particular to enable merging annotations when desired, it is necessary to use a common representation that can accommodate many different kinds of annotation. Therefore, the ANC has chosen to represent all annotations in the LAF dump format. The annotation set for each ANC document includes the header for that document and the primary data with no internal mark-up, together with all applicable annotation documents. The header points to the primary data, as well as each annotation document; annotation documents are linked to the primary data.

The ANC's choice to use the LAF representation makes the data extremely flexible: the primary text can be used with no mark-up or annotations if desired (which is commonly the case for concordance generation, etc.), or the user can choose to deal with a particular annotation set independent of the text (e.g., to generate statistics for POS taggers or parsers). Furthermore, annotations of many different types, or several versions of a single annotation type (e.g., multiple part of speech taggings), can be provided without encountering the problems of incompatibility (in particular, the famous "overlapping hierarchy" problem that arises when different systems assign different boundaries to words or other elements in data). Most importantly, users acquire all annotations in a common format; if users were to generate annotations for the ANC data on their own, each annotation – including annotations of the same type – would be in a different format and require special processing. By rendering all annotations in LAF format, comparison and merging of annotations becomes a far simpler task.

At present, few software systems handle stand-off annotation, and those that do often demand computational expertise beyond what many ANC users – who include linguists, teachers of English as a second language, etc. – have access to. Therefore, the ANC project has developed an easy-to-use tool and user interface (http://americannationalcorpus.org/tools/index.html#xces-parser) (Suderman and Ide, 2006) to merge the stand-off annotations of the user's choice with the primary data and produce the merged document in any of several formats, including, at present, a well-formed XML document in XCES format (suitable for use with various search and access interfaces such as the BNC's XAIRA (http://sourceforge.net/projects/xaira)), WordSmith/MonoConc Pro format, and text with part of speech tags appended to each word and separated by an underscore. The ANC merging tool implements the org.xml.sax.XMLReader, and therefore it is relatively trivial for users to provide their own interface in order to produce output in any format, or to perform other operations on the data (e.g., frequency counts, bigram generation, etc.). By using this tool, the ANC user need never deal directly with or see the underlying representation of the corpus and its stand-off annotations, but gains all the advantages that representation offers.

Because the DCR is still in its development phase, ANC annotation documents do not currently provide RDF/OWL schema mappings to DCR categories. Furthermore, because many ANC annotations are generated automatically using a wide range of freely available or contributed software, determining the mapping for each annotation document may be unfeasible. The ANC will, however, provide the DCR mapping for categories used to annotate its 10 million word "gold standard" sub-corpus, which includes hand-validated annotations for morpho-syntax, syntax, named entities, WordNet senses, and FrameNet frames. As such, the ANC should provide a proof of concept for the LAF architecture, and serve as a usage example upon which others can build.

5 Conclusion

The framework presented here for linguistic annotation is intended to allow for variation in annotation schemes, while at the same time enabling comparison and evaluation, merging of different annotations, and development of common tools for creating and using annotated data. We have developed an abstract model for annotations that is capable of representing the necessary information, while providing a common encoding format that tools can be adapted to manipulate and access, as well as a means to combine and compare annotations. The details presented here provide a look "under the hood" in order to show the flexibility and representational power of the abstract scheme; however, the intention is that annotators and users of syntactic annotation schemes can continue to use their own or other formats with which they are comfortable, and translation into and out of the abstract format will be automatic.

Our framework for linguistic annotation is built around some relatively straightforward ideas: separation of information conveyed by means of structure and information conveyed directly by specification of content categories; development of an abstract format that puts a layer of abstraction between site-specific annotation schemes and standard specifications; and creation of a DCR to provide a reference set of annotation categories. The emergence of XML and related standards, together with RDF/OWL, provides the enabling technology. We are, therefore, at a point where the creation and use of annotated data and concerns about the way it is represented can be treated separately – i.e., researchers can focus on the question of *what* to represent, independent of the question of *how* to represent it. The end result should be greater coherence, consistency, and ease of use and access for linguistically annotated data.

The abstract model that captures the fundamental properties of an annotation scheme provides a conceptual tool for assessing the coherence and consistency of existing schemes and those being developed. The model enforces clear distinctions between implicit and explicit information (e.g., functional relations implied by structural relations in constituent syntactic analyses) and phrasal and functional relations. It is alarmingly common for annotation schemes to represent these different kinds of information in the same way, rendering their distinction computationally intractable (even if they are perfectly understandable by the informed human reader). Hand-developed annotation schemes used in treebanks are often described informally in guidebooks for annotators, leaving considerable room for variation; for example, (Charniak, 1996), notes that the PTB implicitly contains more than 10,000 context-free rules, most of which are used only once. Comparison and transduction of schemes becomes virtually impossible under such circumstances. While requiring that annotators make relations explicit and consider the mapping to the abstract format increases overhead, we feel that the exercise will help avoid such problems, and can only lead to greater coherence, consistency, and inter-operability among annotation schemes.

Notes

1. In fact, the term "primary resource" is somewhat misleading, since each transcription or annotation level can be regarded as a primary resource for another level. This notion of multiple information layers is the underlying principle for stand-off mark-up.

2. Participants: Nuria Bel (Universitat de Barcelona), David Durand (Brown University), Henry Thompson (University of Edinburgh), Koiti Hasida (AIST Tokyo), Eric De La Clergerie (INRIA), Lionel Clement (INRIA), Laurent Romary (LORIA), Nancy Ide (Vassar College), Kiyong Lee (Korea University), Keith Suderman (Vassar College), Aswani Kumar (LORIA), Chris Laprun (NIST), Thierry Declerck (DFKI), Jean Carletta (University of Edinburgh), Michael Strube (European Media Laboratory), Hamish Cunningham (University of Sheffield), Tomaz Erjavec (Institute Jozef Stefan), Hennie Brugman (Max-Planck-Institut für Psycholinguistik), Fabio Vitali (Universite di Bologna), Key-Sun Choi (Korterm), Jean-Michel Borde (Digital Visual), and Eric Kow (LORIA).

3. Draft documents and working papers for the various areas, including morpho-syntactic annotation (ISO/TC 37/SC 4 document N225), syntactic annotation (ISO/TC 37/SC 4 document N244), word segmentation (ISO/TC 37/SC 4 document N233), etc. are available at http://www.tc37sc4.org/.

4. In these examples, *number* is given literal values. However, with OWL it is possible to restrict the range of possible values by enumeration.

5. It is of course possible to generate a LAF representation without making implicit information explicit, thus placing the burden of extracting the information on the user of the LAF instantiation. LAF guidelines can "require" explicitness in principle, but they cannot ensure that it is enforced.

6. A full description of the XML feature structure representation can be found in ISO standard 24610-1. See also the TEI guidelines, Chapter 16 (http://www.tei-c.org/release/doc/tei-p5-doc/html/FS.html).

7. The use of <`feat`> elements in this example shows the use of a simplified XML format for feature structures that is sufficient for many types of annotation information. In cases where the full power of FS representation is required, the TEI/ISO standard XML representation for feature structures can be used.

References

Bird, S. and Liberman, M. (2001). A Formal Framework for Linguistic Annotation. *Speech Communication*, 33(1–2):23–60.

Carroll, J., Minnen, G., and Briscoe, T. (2004). Parser Evaluation. In Abeillé, A., editor, *Treebanks: Building and Using Parsed Corpora*, pages 299–316, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Charniak, E. (1996). Treebank Grammars. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*, pages 1031–1036, MIT Press.

Cole, R., Mariani, J., Uszkoreit, H., Zaenen, A., and Zue, V., editors (1997). *Survey of the State of the Art in Human Language Technology*. Cambridge University Press, first edition.

Ide, N. and Romary, L. (2001). A Common Framework for Syntactic Annotation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 298–305, Toulouse, France.

Ide, N. and Romary, L. (2002). Standards for Language Resources. In *Proceedings of the Third Language Resources and Evaluation Conference (LREC)*, pages 839–844, Las Palmas, Gran Canaria, Spain.

Ide, N. and Romary, L. (2004a). A Registry of Standard Data Categories for Linguistic Annotation. In *Proceedings of the Fourth International Language Resources and Evaluation Conference (LREC)*, pages 135–139, Lisbon, Portugal.

Ide, N. and Romary, L. (2004b). International Standard for a Linguistic Annotation Framework. *Natural Language Engineering*, 10(3–4):211–225.

Suderman, K. and Ide, N. (2006). Layering and Merging Linguistic Annotations. In *Proceedings of the Fifth Workshop on NLP and XML (NLPXML)*, pages 89–92, Trento, Italy.