The XML Framework and Its Implications for Corpus Access and Use

Nancy Ide

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

Abstract

The eXtensible Markup Language (XML) (Bray, et al., 1998) is the emerging standard for data representation and exchange on the World Wide Web. The XML Framework includes very powerful mechanisms for accessing and manipulating XML documents that are likely to significantly impact the way annotated corpora are created and accessed. This paper outlines a few of the possibilities.

Introduction

The eXtensible Markup Language (XML) (Bray, et al., 1998) is the emerging standard for data representation and exchange on the World Wide Web. At its most basic level XML is a document markup language directly derived from SGML (i.e., allowing tagged text (elements), element nesting, and element references). As such, translation of an SGML encoded document into XML is relatively trivial. However, various features and extensions of XML make it a far more powerful tool for data representation and access than SGML. The following outlines some of these mechanisms and suggests ways in which they can be used for creation and exploitation of annotated corpora.

XML links

The recommended practice in encoding annotated corpora is to maintain all or most annotations in separate documents, each of which references appropriate locations in the document containing the original data (Ide & Brew, 2000). This strategy yields, in essence, a finely linked hypertext format where the links specify a semantic role rather than navigational options. That is, links signify the location(s) where markup contained in a given annotation would appear in the document to which it is linked. As such, annotation information comprises remote or “stand-off” markup that is virtually added to the document to which it is linked. In principle, the original data could contain no markup at all (or, more likely, markup for gross logical structure only); all markup could be retained in separate documents with links into the original based on offsets.

The standoff scheme, then, requires addressing XML elements, as well as characters and chains of characters within those elements. It also requires that elements and characters can be addressed both within the same document and in other XML documents. XML provides the following linking mechanisms, which are substantially more powerful than the mechanisms provided in SGML, which satisfy these requirements:

- XLink (DeRose, et al., 2000), a mechanism for specifying a link (uni-directional or more complex linking structures) between two or more resources or portions of resources;
- the XML Path Language (XPath) (Clark & DeRose, 1999), an extended addressing syntax that defines a concise notation for element localization in the document tree (as defined by the nesting of elements in the document itself), and allows addressing text fragments within a particular element by providing predicates for manipulating chains of characters;
- XPointer (DeRose, Daniel, & Maler, 1999), which extends XPath syntax to allow addressing points and ranges as well as nodes, locating information by string matching, and use of addressing expressions in URI-identifiers as fragment identifiers.

For example, the Xpath expression /div/p[2]/s[3] specifies the third <s> (sentence) element within the second <p> (paragraph) element within each <div> (text division) element; /descendant::p specifies all <p> elements in the document. In addition, XPath allows addressing text fragments within a particular element by providing predicates for manipulating chains of characters. The expression substring(/p/s[2]/text(), 6) selects the string "one would expect that the whole sky would be as bright as the sun, even at night." from the following text:

<p><s id="d3p13s4">The difficulty is that in an infinite static universe nearly every line of sight would end on the surface of a star.</s></p>

The expression substring(/p/s[2]/text(), 10, 12) selects "would expect". Thus the reference is made by specifying (1) the address (absolute or relative) of the element closest to the substring to be referred to, and (2) the substring within this element.

The Xlink mechanism can be used to link corresponding segments of two or more primary texts (As for alignment of text or speech), or to link annotation documents to a base document containing the primary text. For example, in the following, annotation information (e.g., morpho-syntactic information) about a specific token (<tok>) is linked to the string of characters in the original text to which it applies:

<tok
   xlink:href="substring(/p/s[2]/text(),10,12)"
>

Although this example shows linkage for text, the mechanism provides for linking resources in any medium (audio, video, etc.), which allows for linking speech,
external images, video, applets, form-processing programs, style sheets, etc.

In addition to specifying the target location for information in the same or external documents, XLink attributes can be used to specify the role of the link, i.e., how the link should be activated (by hand, or automatically by the browser) and what to do with the target fragment (replace it or insert it into the source document).

In XML, annotated fragments are referenced by the URI (remote or local) of the target resource, and an extended pointer identifying a element and, where necessary, the selected substring of that element’s content, as in the following:

```xml
<tok xlink:href="http://www.loria.fr/doc.xml#xptr(substring(/p/s[2]/text(), 10, 12))">
  Annotation resulting from automatic processing (marking of sentence boundaries, tokens, links between parallel texts, etc.) often includes thousands of links to the same external document. Repetition of the document name on every relevant element in an annotation document would obviously significantly multiply its size. XML includes an attribute `xml:base` (Marsh, 2000) that can be used to specify inheritance of an attribute. For example, in the following text:

```xml
<chunk xml:base="http://www.mysite.edu/doc.xml#">
  <tok xlink:href="xptr(substring(/p[s[2]/text(), 10, 12))">
  </chunk>
```

the value of the attribute `xml:base` specified on the `<chunk>` element is inherited by the two `<tok>` elements that are its children, and therefore need not be re-specified. The inclusion of `xml:base` in the XML specification ensures that conformant XML processors will handle it (unlike SGML).

**XML transformations**

The Extensible Style Language (XSL) is a part of the XML framework, consisting of two parts: the best known is the XSL formatting or "style sheet" language; and a powerful tree-traversal language, XSLT (Clark, 1999), that can be used to convert any XML document into another document in any form (e.g., XML, well-formed HTML, plain text, etc.) by selecting, rearranging, and/or adding information to it. The transformed documents may or may not be intended for rendering data on a computer screen, but may be used simply to move data from one computer system or program to another (e.g., to transduce between encoding and/or annotation formats, etc.).

XSLT supports the following kinds of document manipulation:

- selection of elements or portions of element content using the XPath syntax;
- rearrangement or transformation of extracted information (including not only text content but also element names, etc.) in the target document;
- addition of information in the target document.

A suite of documents representing a base text (or texts) and its annotations can be manipulated to serve any application that relies on part or all of its contents. Thus, XSLT is likely to have the most to offer for manipulation of and access to annotated corpora.

XSLT is relatively complex and will not be described in detail here. A short example can provide some idea of the possibilities. Using as input a document containing morpho-syntactic information (e.g., a document containing the fragment in Figure 1), the XSLT document in Figure 2 can be used to create an HTML document that displays a text in "word | lemma | pos" form. When the resulting HTML document is loaded into a browser, it will display the following:

```html
It|it|PPER3 was|be|PAST3 a|a|DINT bright|bright|ADJE cold|cold|ADJE day|day|NNL...
```

Figure 1 : Fragment of a cesAna document

1 Full documentation is available at http://www.w3.org/TR/xslt.
2 Note that this document, encoded according to the xcesAna specifications (Ide, Bonhomme, & Romary, 2000), contains full smentation and annotation information, including full morpho-syntactic specifications for all potential annotations and the results of automatic disambiguation.
XML Schemas

The XML Schema definition language (Thompson, et al., 2000; Biron & Malhotra, 2000) enables document creators to constrain and document the meaning, usage and relationships of the constituent parts of XML documents: datatypes, elements and their content, and attributes and their values. Schemas can also be used to provide default values for attributes and elements. As such, XML schemas provide means to define an abstract data model for a class of documents. While duplicating (or making explicit) some of the capabilities provided by XML DTDs, they significantly extend their power and provide for much tighter validation of document form and content.

XML schemas have considerable implications for development of annotated corpora. The following lists only a few possibilities for the application of XML schemas:

- different attribute declarations and/or content models can apply to elements with the same name in different contexts. This allows for more tightly constrained content models than possible with DTDs. For example, names in headers (names of authors, etc., consisting of the usual "first name", "last name" elements) and names in the text ("named entities") should have

---

3 See also Erjavec et al. (2000)
Conclusion

This paper outlines some of the potential uses of the mechanisms provided within the XML framework for the creation and use of annotated corpora. As the XML standard develops, more ways to exploit XML will no doubt emerge. For instance, it is likely that corpora and their annotations will be distributed over servers throughout the world in the future, rather than being maintained at a single site; XML is explicitly designed to handle distributed data. Also, XML’s provision for accessing part or all of multiple DTDs from a single document provides an elegant means to represent and manipulate standoff annotation documents.

XML is an international standard that will become the base of information exchange and access over the World Wide Web. As we develop annotated corpora that are intended to be maximally usable and reusable for multiple applications, it makes sense to exploit the XML framework to ensure the compatibility and flexibility our applications (and future applications we cannot yet envision) require.

Acknowledgements

The author gratefully acknowledges Laurent Romary and Patrice Bonhomme for their contributions to the ideas in this paper, and for providing the examples of linkage and the use of XML schemas.

References


