Layering and Merging Linguistic Annotations

Keith Suderman
Department of Computer Science
Vassar College
Poughkeepsie, NY USA
suderman@cs.vassar.edu

Nancy Ide
Department of Computer Science
Vassar College
Poughkeepsie, NY USA
ide@cs.vassar.edu

Abstract

The American National Corpus and its annotations are represented in a stand-off XML format compliant with the specifications of ISO TC37 SC4 WG1's Linguistic Annotation Framework. Because few systems that enable search and access of the corpus currently support stand-off markup, the project has developed a SAX *like* parser that generates ANC data with annotations in-line, in a variety of output formats.

1 Introduction

The American National Corpus (ANC) project[1] recently released its 2nd release consisting of approximately 22 million words of data, representing a variety of genres of both written and spoken data. The corpus is annotated with several layers of automatically produced linguistic information, including sentence and token boundaries, part of speech using two different POS tagsets (a version of the Penn tagset[2] and the Biber tagset[3]), and noun chunks and verb chunks.

ANC primary documents are plain text (UTF-16) documents and are treated as "read only" resources. All annotations are represented in stand-off XML documents referencing spans in the primary data or other annotation documents, using the XCES[4] implementation of the specifications of ISO TC37 SC4's Linguistic Annotation Framework (LAF) (Ide and Romary, 2004). Because few systems that enable search and access of the corpus currently support stand-off markup, the project has developed a parser that generates ANC data with annotations in-line, in a variety of output formats.

This demonstration will show the "life-cycle" of an ANC document, from acquisition of a document in any of a variety of formats (MS Word, PDF, HTML, etc.) through annotation and final representation in the stand-off format. The ANC tool for merging annotations of the user's choice with the primary data to produce a single document with in-line annotations will also be demonstrated.

2 ANC Document Life-Cycle

Documents to be included in the ANC are acquired in many different formats, including MS Word, PDF, HTML, Quark Express, etc. Processing involves a series of steps, which are outlined below.

2.1 Conversion from original format to "rudimentary" XML

The ANC receives documents in a variety of different formats. The first step in processing is to convert the input documents into XCES XML with basic structural annotations included. The most common types of file formats encountered are:

- **Microsoft Word.** The release of OpenOffice 2 has greatly simplified the processing of MS Word documents. OpenOffice uses XSL and XSLT stylesheets to export files to XML and ships with stylesheets to generate DocBook and TEI-compliant formats. We modified the TEI stylesheet to create XCES XML. OpenOffice's Java API enables us to automate and integrate OpenOffice with later processing steps.
- **XML/SGML/HTML**. processing of XML files typically involves using XSLT to map element names to XCES. SGML and HTML files typically require preprocessing to render them into valid XML, followed by the application of an XSLT stylesheet to convert them to XCES.

[1] http:// americannationalcorpus.org

[2] http://americannationalcorpus.org/FirstRelease/gatetags.txt

[3] http://americannationalcorpus.org/FirstRelease/Biber-tags.txt

[4] http://www.xces.org

- **Quark Express.** Several publishers provided documents prepared for publication using Quark Express. Quark documents can be exported in XML, but doing so is worthwhile only if the creator of the document takes advantage of Quark's style-definition facilities (which was not the case for any of the contributed Quark documents). We therefore exported the documents in RTF; however, many fonts and special characters are improperly rendered, and fairly extensive manual editing was therefore required to render the files into a format that could be used. Once edited, the same procedures for MS Word documents are used to generate XCES.
- **PDF.** Bitmap PDF files are unusable for our purposes. Adobe Acrobat can generate plain text from PDF, although this process loses much of the formatting information that would be desirable to retain to facilitate later processing. In some cases, ligatures and other special characters are improperly represented in the text version, and it is not always possible to automatically detect and convert them to conform to the original. PDF documents with two or more columns cannot, to our knowledge, be extracted without some misordering of the text in the results.
- **Other formats.** Other formats in which the ANC has acquired documents include plain text and plain text that employed a variety of proprietary markup languages. These documents are processed on a case by case basis, using specialized scripts.

2.2 GATE processing and annotation

We use the University of Sheffield's GATE system[5] for the bulk of ANC document processing and annotation, currently including tokenization, sentence splitting, part of speech tagging, noun chunking, and verb chunking. Most annotations are produced using GATE's built-in ANNIE components; we have, however, modified the ANNIE sentence splitter and created several Java plug-ins for use in GATE that perform basic bookkeeping, renaming of annotations/features, moving of annotations between annotation sets etc. We have also developed a scripting language (XORO[6]) for use with GATE to enable easy bulk processing and re-processing of the entire corpus, or to apply selected annotation steps without having to load the files into a GATE corpus or data store. This eases iterative development as documents are added and tools are refined.

2.3 Creation of standoff annotation documents

We have developed several custom processing resources that plug into GATE to generate standoff annotations in the XCES implementation of the LAF format. The last step in our GATE pipeline is to create the primary text document and generate all the required standoff annotation files.

3 Standoff Format

The ANC standoff format for annotations is a simple graph representation, consisting of one node set and one, or more, edge sets. The node set is represented by the text itself, with an implied node between each character. Each edge set is represented by an XML document and may contain one or more annotation types: logical structure, sentence boundaries, tokens, etc.

An ANC header file for each document is used to associate the source text with the standoff annotation documents; for example:

```
<cesHeader>
  ...
  <annotations>
    <annotation type="content"
       ann.loc="en_4065.txt">
       Text content</annotation>
    <annotation type="logical"
       ann.loc="en_4065-logical.xml">
       Logical structure</annotation>
     <annotation type="s"
       ann.loc="en_4065-s.xml">
       Sentence boundaries</annotation>
     <annotation type="hepple"
       ann.loc="en_4065-hepple.xml">
       Hepple POS tags</annotation>
     <annotation type="biber"
       ann.loc="en_4065-biber.xml">
       Biber POS tags</annotation>
     <annotation type="vp"
       ann.loc="en_4065-vp.xml">
       Verb chunks</annotation>
     <annotation type="np"
       ann.loc="en_4065-np.xml">
       Noun chunks</annotation>
  </annotations>
       ...
</cesHeader>
```

ANC annotation documents are marked up with the XCES representation of the nodes and edge sets of the annotation graph. The following shows a segment of the document containing part of speech annotation:

[5] http://gate.ac.uk

[6] http:// americannationalcorpus.org/xoro.html

```
<cesAna
   xmlns="http://www.xces.org/schema/2003"
   version="1.0.4">
   <struct type="tok" from="4" to="6">
      <feat name="base" value="in"/>
      <feat name="msd" value="IN"/>
   </struct>
   <struct type="tok" from="7" to="11">
      <feat name="msd" value="DT"/>
      <feat name="base" value="this"/>
   </struct>
   <struct type="tok" from="12" to="19">
      <feat name="base" value="chapter"/>
      <feat name="msd" value="NN"/>
   </struct>
   ...
</cesAna>
```

Each `<struct>` element represents an edge in the graph; values of the *from* and *to* attributes denote the nodes (between characters in the primary text document) over which the edge spans.

3.1 Annotating discontiguous spans

Presently, the ANC includes standoff annotations that reference contiguous spans of data in the original (primary) document. However, we plan to add a wide variety of automatically-produced annotations for various linguistic phenomena to the ANC data, some of which will reference discontiguous regions of the primary data, or may reference annotations contained in other standoff documents. This is handled as follows: given an annotation graph, *G*, we create an edge graph *G'* whose nodes can themselves be annotated, thereby allowing for edges between the edges of the original annotation graph *G*.

For example, consider the sentence "My dog has fleas." The standoff annotations for the tokens would be:

```
                    1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6
|M|y| |d|o|g| |h|a|s| |f|l|e|a|s|

<struct … id="t1" from="0" to="2"/>
<struct … id="t2" from="3" to="6"/>
<struct … id="t3" from="7" to="10"/>
<struct … id="t4" from="11" to="16"/>
```

Now consider the dependency tree generated by Minipar[7] given in Figure 2. The tree can be represented by annotating the token elements in the standoff annotation document as follows:

```
<!-- Define some pseudo nodes -->
<node type="root" id"E0" ref="t3"/>
<node type="clone" id="E2" ref="t2"/>


<!-- Define edges in dependency tree -->
<struct type="subj" id="s1"
        from="t3" to="E2"/>
<struct type="s" id="s2"
        from="t3" to="t2"/>
<struct type="gen" id="gen"
        from="t2" to="t1"/>
<struct type="obj" id="obj"
        from="t3" to="t4"/>
```

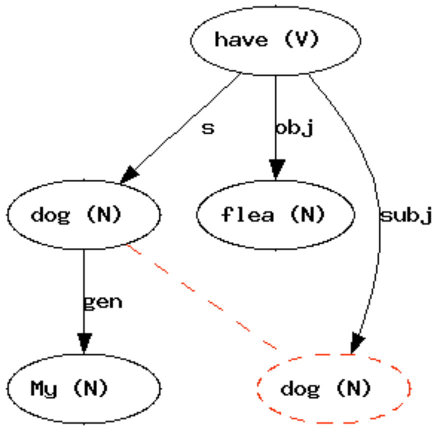

Figure 2. Dependency tree generated by Minipar.[8]

4 Creating In-line Annotation Documents

We have developed an "XCES Parser"[9] that implements the org.xml.sax.XMLReader interface to create ANC documents containing in-line annotations in XML (or any other format).

The XCES parser works as follows: annotations to be loaded are selected with the `org.xml.sax.XMLReader.setProperty()` method. The selected annotation sets are then loaded into a single list in memory and sorted, first by offset and, if the offsets are the same, secondly by annotation type. At present, the ordering of annotation types are hard coded into the parser; work is underway to make the XCES parser "schema aware" so that embedding specifications can be provided by the user. Once the text is loaded and sorted, the appropriate SAX2 events are generated and dispatched to the org.xml.sax.ContentHandler (if one has been registered with the parser) in sequence to simulate the parsing of an XML document. While the parser will allow the programmer to specify an ErrorHandler, DTDHandler, or EntityResolver, at this time no methods from those interfaces will be invoked during parsing.

In the current version of the XCES parser, when overlapping annotations are encountered, they are "truncated". For example:

```
<s>Sentence <em>one.</s><s>Sentence</em>
two.</s>
```

[7] http://www.cs.ualberta.ca/~lindek/minipar.htm

[8] Image generated by http://ai.stanford.edu/~rion/parsing/minipar_viz.html

[9] http://americannationalcorpus.org/tools/index.html#xces-parser

becomes

```
<s>Sentence <em>one.</em></s><s>Sentence
two.</s>
```

Work is underway to provide the option to generate milestones in CLIX/HORSE (DeRose, 2004) format to represent overlapping hierarchies.

4.1 Using the XCES parser

The XCES parser can be used in three ways:

- from the command line. The xces-parser.jar file can be run as a command line program to print XML with inline annotation to standard output.
- as the XML parser used by other applications. For example, Saxon[10] can take the name of the parser to use to parse the source document as a command line parameter. This allows us to apply XSLT stylesheets to ANC documents without having to translate them into XML first.
- as a library for use in other Java applications. For example, The ANC Tool[11] is a graphical front end to the XCES parser.

4.2 The ANC tool

The ANC Tool provides a graphical user interface for the XCES parser and is used to convert ANC documents to other formats. Users specify their choice of annotations to be included. Currently, the ANC Tool can be used to generate the following output formats:

- XML XCES format, suitable for use with the BNC's XAIRA[12] search and access interface;
- Text with part of speech tags appended to each word and separated by an underscore;
- WordSmith/MonoConc Pro format.

The ANC Tool uses multiple implementations of the `org.xml.sax.DocumentHandler` interface, one for each output format, which the XCES parser uses to generate the desired output. Additional output formats can be easily generated by implementing additional interfaces.

Of course, if the target application understands annotation graphs, there is no need to bother with the XCES parser or conversion to XML. For example, we provide several resources for GATE that permit GATE to open and read ANC documents with standoff annotations, or to load standoff annotations into an already loaded document.

5 Future Work

Currently the XCES parser is a *proof of concept* rather than a production grade tool. The parser is being augmented to invoke all the appropriate methods from the `org.xml.sax.*Handler` interfaces and throw the proper `SAXExceptions` at the appropriate times. We are also providing for some level of SAX conformance, rather than simply "doing what Xerces does".

6 Conclusion

The ANC has implemented an efficient pipeline for the processing of text into a corpus of machine usable documents. For some document types this process is almost completely automated and can be regarded as a *Corpus-Builder-in-a Box*: raw data goes in one end, and a fully annotated corpus with standoff annotations comes out the other.

The use of standoff annotations allows for an accurate representation of the ANC data as provided by the contributors and allows us to easily provide several modular annotation sets that can be included or excluded by the end user as desired. By providing a SAX like parser for ANC documents, we are able to leverage a number of available XML tools without the restrictions imposed by an XML representation of the documents. For users who are not interested in XML or standoff annotations, the plain text version is preserved.

References

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

Ide, N., Romary, L. (2004). International standard for a linguistic annotation framework. *Journal of Natural Language Engineering*, 10:3-4, 211-225.

[10] http://saxon.sourceforge.net/

[11] http:// americannationalcorpus.org/tools/anctool.html

[12] http://sourceforge.net/projects/xaira