Annotation Science

Representing Language Resources and Their Annotations

Nancy Ide
Department of Computer Science
Vassar College
USA

The need to "understand" texts depends on the recognition of linguistic features.
We gather large bodies of language data (text, speech) and annotate it for a wide range of linguistic phenomena.
Used to train automatic annotation software, study.

The "science" of annotation involves:
- How to gather data of appropriate types
- How to decide which types of data are needed
- How to semi-automate the process of hand annotation
- How to validate annotators' work
- How to represent the data and its annotations to make it most (re)-usable

Increasingly, data is annotated for more than one type of linguistic phenomenon:
- Morpho-syntax, syntax, word senses, semantic roles, discourse structure, entities, events...
This leads to the need to enable merging annotation information of different types.
Also, many different annotation schemes are used for the same phenomenon.
This leads to the need to enable merging annotation information of the same type.
The problems of combining information from WordNet, FrameNet, VerbNet, etc. are a small example.

A major aspect of the science of annotation is developing standards, to enable merging and comparison of different annotations:
- Standards for content categories
- Standards for representation

There have been many standardization efforts, none yet fully successful:
- TEI and the Corpus Encoding Standard (representation)
- EAGLES/ISLE (content)
- De facto standards such as WordNet for sense tags

Very often no separation between:
- Annotation content categories
  - E.g., sense tags, syntactic categories, POS categories, etc.
- Annotation structure
  - Relations among parts of the annotation

Essential to separate these two.
Content Categories

• Most people think only of this when considering annotation standards
• Very difficult to standardize
• Problems:
  - Different theoretical approaches
  - Different levels of granularity
• Many efforts over the past 15 years
  - EAGLES/ISLE
  - For the most part we use de facto standards

Annotation Structure

• Standardization of annotation representation has focused on formats (e.g. XML) rather than representing the relevant relations among content categories
• Rather than a prescribed format (e.g. XML) we need to focus on means to
  - Represent any annotation type
  - Enable trivial mapping of one format to another

Why Is This Difficult?

• Because annotation content is often not distinguished from annotation structure, some information may be conveyed implicitly by the structure
  - This can make it difficult (or impossible) to extract the information
• Example:
  - Use same structural mechanism to convey different relations
    • E.g., nesting in LISP-like format:
      - A set of alternatives?
      - An ordered list?
    • Only way to know is to understand the content categories!

(X ((Y) (Z)))

Other problems

• Inconsistency
  - Some information represented explicitly, some represented via structure
  - Example: Syntactic Annotation

Carroll, Minnen, and Briscoe

What's Going On?

Category information: 'NP'
Relational information: 'SBJ'
(implicit head)
Node identification: 1
Category information: 'VP'
Relational information: 'xcomp'
(implicit head)
Node reference *-1

subj(intend,Paul,_) xcomp(intend,leave,_,to) subj(leave,Paul) dobj(leave,IBM,_)
What are the Underlying Principles?

- These annotation schemes include:
  - Relational information
    - Relations implicit in the hierarchy of bracketed components
    - Explicitly specified relations
  - Category information
    - Syntactic category (noun phrase, prepositional phrase...)
    - Role (subject, object...)
    - Thematic role (agent, patient...)
- Inconsistent!
  - This makes it difficult to extract the information or map to another scheme via automatic means.

Another Problem

- Often, annotations are in the same document/file as the primary data.
- Hard to have several different types of annotation for the same data
- Hard/impossible to have alternative annotations of same type
- Hard to maintain
- Disrupts text, may lose information (e.g., punctuation, spacing)

Why is this a bad idea?

- Hard to have several different types of annotation for the same data
- Hard/impossible to have alternative annotations of same type
- Hard to maintain
- Disrupts text, may lose information (e.g., punctuation, spacing)

Stand-off Annotation

- Annotations in separate documents, linked to original
- Use of stand-off annotations is becoming the standard practice
- But:
  - Which annotations go in separate annotation documents, which in the same annotation document?
  - Can annotations be linked not to the original data, but rather to other annotation documents?

ISO TC37 SC4

- International Standards Organization group for Language Resource Management
- Addressing all these issues
- WG1: Linguistic Annotation Framework (LAF)
- Other work groups for specific annotation types
**LAF Approach**

- Develop a common, abstract model that can capture all types of annotation information, regardless of the physical encoding
- Develop a generic, XML instantiation of the model, to and from which specific formats can be mapped
- Define a common set of data categories, for reference and use by annotators

**Principles**

- Separation of data and annotations
  - Stand-off annotation
- Separation of user annotation formats and the exchange (“dump”) format
  - Mappable to one another
- Separation of referential structure and annotation content in dump format
- Separation of annotation structure (relationships among parts) and content (data categories) in representation of annotations

**Abstract Model**

- Annotations represented as a graph of feature structures
  - Nodes are locations in primary data or other annotations (“edge graph”)
  - Edges are labeled with feature structures containing the annotation content
- Any format instantiating the model can be trivially mapped to another format
- Map to a “pivot format” (XML) that instantiates the model

**Main Idea**

[Diagram showing the main idea with data flow from Format A through Pivot Format to Format B, with intermediate formats A1, A2, and A3]

**Stand-off Annotation**

- Language data is regarded as “read-only” and contains no annotations
- All annotations are contained in stand-off documents linked to the primary data or other annotation documents
User formats

- Users may use any format for annotations
- Requirement: user’s annotation format is automatically mappable to the feature structure-based data model instantiated by the pivot format
  - Annotation information in the original format must be made explicit in the pivot format representation
- In principle, users will never deal directly with, or even see, the pivot format

Data Category Registry

- Addresses issue of standardization of annotation content
- Overall approach:
  - Proceed cautiously!
  - Initially, implement categories that are widely used and relatively low-level

Functions of DCR

- Provide a precise semantics for annotation categories
  - can be used “off the shelf” or modified to serve specific needs
- Provide a set of reference categories onto which scheme-specific names can be mapped
- Provide a point of departure for definition of variant, more precise, or entirely new data categories

Reference Point

- DCR provides a set of reference concepts
- Annotator provides a Data Category Specification (DCS)
  - mapping between scheme-specific instantiations and concepts in the DCR
  - Including differences, departures
  - Provides documentation for the user’s annotation scheme
- DCS included or referenced in data exchange
  - provides receiver with information to interpret annotation content or map to another instantiation
  - semantic integrity guaranteed by mutual

How Do I Design an Encoding Scheme for Annotations?

- Short answer:
  - Any way you want, provided you follow a coherent data model

Desiderata for an Annotation Scheme

- captures the features you need
- easily processed at your site as well as at others
- compatible with other schemes, to enable merging, comparison, etc.
- makes the annotations reusable and extensible
Step One: From Theory to Model

- Figure out the linguistic categories and features you want to represent without regard for the issue of physical format
  - E.g.
    - Part of speech: how detailed is the category information? Lemma specified? Etc.
    - Syntax: what are the categories (NP, VP)? Is syntactic role (subject, ...

Step Two: From Model to Representation

- Develop a way to represent the information
  - Preferably in XML, but not necessarily
  - Processable by your software
  - No information loss
  - As long as the information is there, no need to worry about specific XML format

Example (Part of Speech)

- Lots of options:

  `<w cat="NNP" lemma="dog">dogs</w>`

  `<seg target="xptr(substring(/p/s[1]/text(),1,5))" lex="MyCats#DX51"/>

Why is the Exact Format Unimportant?

- It is easy to transduce one format to another
  - ...as long as the information is there and mapping is one-to-one
- Annotators can retain their own internal formats
- Must be built on a common data model

LAF Development

- LAF has gone through a slow evolution
  - Model development
  - Consideration of processing needs
  - Application to different annotation types/structures/formats
  - Adjustments to development in other ISO TC37 SC4 Working Groups on specific annotation types and feature structures
  - “Proof of concept” instantiation in the American National Corpus

Basic Model

- Annotation content represented by feature structures
  - Powerful means to represent any/all annotations
- Referential structure represented as a directed acyclic graph (DAG)
  - Enables exploitation of well-understood graph traversal and
Referential Structure

• Means by which annotation content is associated with primary data or other annotations
  - Very simple DAG model
  - No need to consider internal structure of annotation content (i.e. relations among bits of annotation information)

Primary Data

• Primary data contains no annotations
  - "Read-only"
  - Modifications can be regarded as annotations
• Insistence on the existence of a primary segmentation of the primary data
  - Identifies contiguous sequences of indivisible logical units
    • For text, usually a character

Primary Segmentation

• Set of disjoint edges over primary data
  - Vertices
    • Virtual, located between each logical unit S
    • Sequentially numbered
  - Edges
    • Each edge (x,y) in the graph delimits a non-divisible region of primary data
• Multiple primary segmentations may be defined over a single primary data set
  - Specify segmentations at different levels of granularity
  - A segmentation is "primary" vis a vis a given annotation, not the data itself
• Edges in a primary segmentation can be defined over any span of contiguous primary data, regardless of its length
  - For text, most common primary segmentation is the token

Primary segmentation with edges (0,3), (4,9), and (10,16)

| T|h|e| |c|l|o|c|k| |s|t|r|u|c|k| |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Referring to Primary Segmentation

• Define an edge graph over the edges in the primary segmentation
  - Given an edge set, $E$, create an edge graph $E'$ such that for each edge $(x,y)$ in $E$, there is a vertex $xy$ in $E'$$
• Annotations are associated with regions of primary data by referencing the edge graph vertices
  - Annotations never reference the primary data directly
• Edges in $E'$ are defined when annotations reference vertices in $E'$
  - Vertices may or may not be contiguous
• An annotation is associated with vertices in $E'$ as follows:
  1. Create a new vertex, $v$
  2. Label it with the FS containing the annotation content
  3. Create an edge from $v$ to 0 or more vertices in $E'$
    - Zero reference is used in the special case where the annotation applies to information not present in the data
• References to 2 or more vertices in $E'$ by by

As many annotations as desired can reference the same segmentation or be layered over lower-level annotations

Annotating Annotations
• An annotation may be referenced from other annotations
  1. Create a new vertex, $v'$
  2. Create an edge from $v'$ to one or more vertices associated with an annotation
  3. Label the edge with the FS containing the annotation content
• The strategy described above may be applied recursively, thus creating a DAG whose leaves are the vertices in $E'$

XML Instantiation
<!-- edges over primary data -->
<edge id="e1" from="0" to="3"/>
<edge id="e2" from="4" to="9"/>
<edge id="e3" from="10" to="16"/>
<edge id="e4" from="17" to="23"/>
<edge id="e5" from="23" to="27"/>
Token Annotation

<edge id="t2" targets="e2">
  <fs type="token">
    <f name="base" sVal="clock"/>
    <f name="pos" sVal="NN"/>
  </fs>
</edge>

NP Annotation

<edge id="npl" targets="t1 t2">
  <fs type="NP">
    <f name="number" sVal="singular"/>
  </fs>
</edge>

Multiple Targets

- When referring to annotations, edge targets typically represent components
  - E.g. in the example: "the" and "clock" are (sequential) components of "NP"
- But this is not always the case
  - Could be e.g. a list of co-referents
- Solution: let the processor deal with it using the FS type

Feature Structures

- Each edge is labeled with a feature value
  - Can be another feature structure, collection (list, bag, set), atom
- Alternation and grouping handled by the FS mechanisms
- Have a set of "basic" FS mechanisms
  - 90% of annotations use only these
  - Annotations may (optionally) use only this set
- Ease of use
  - No need to implement procedures to handle full power of FS

Summary

- The science of annotation is now recognized as more difficult than was thought ten years ago
- Advances in technology have enabled more rapid and thoughtful development in recent years
- Moving toward harmonization of annotation practices
American National Corpus

Nancy Ide
Department of computer science
Vassar College

Why American English?

• Notorious syntactic differences between British and American English have plagued developers of linguistic resources and statistical NLP tools
  – COMLEX
  – FrameNet
• See Fillmore et al., LREC 1998

British vs. American English

• Lexical Items
  • Bobby vs. cop, underground vs. subway, lorry vs. truck, pavement vs. sidewalk, football vs. soccer...
• Grammatical structures
  • "She could not endure to live with him" vs. "She could not endure living with him."
  • "Have you a pen?" vs. "Do you have a pen?"
• Modals
  • "shall" vs. "will", "should" vs. "ought", "would" vs. "should"
• Adverbial Usage
  • "Immediately I get home" vs. "As soon as I get home"
• Support Verbs
  • "take a decision" vs. "make a decision"

Why we need an ANC

• Brown Corpus of American English
  – Too small to provide representative examples
  – Pre-1960 only
  – No spoken data
• British National Corpus
  – Not representative of American English
  – Texts up to 1993 only
• Wall Street Journal corpus
  – Domain specific language use

Why not the web?

• Unlike the “web as corpus”, the ANC
  – Contains only American English
  – Represents multiple genres
    • vs. the Wall Street Journal corpus widely used in NLP research
  – Can be annotated and re-distributed
  – Has already been cleaned up and marked up

ANC Background

• June 1998
  – ANC proposed at LREC'98 by Charles Fillmore, Nancy Ide, Daniel Jurafsky, Catherine Macleod
• May 1998
  – Publisher's Day in Berkeley in conjunction with DSNA
• November 1999
  – Project launched
• October 2003
  – First release of 12 million words
• December 2005
  – Second Release of 22 million words
Contributors

• “Founding” consortium members
  – $21,000 over 3 years
  – Contribute texts
• Linguistic Data Consortium
  – Licensing and distribution of the ANC
• Vassar and Northern Arizona University
  – Expertise and manpower for corpus creation and annotation

ANC Consortium

Pearson Education
Random House Publishers
Langenscheidt Publishing Group
Harper Collins Publishers
Cambridge University Press
Bloomsbury Press
Microsoft Corporation
Shogakukan,Inc.
Associated Liberal Creators Press
Taishukan Publishers
Kankyusha Publishers
Oxford University Press
IBM Corporation

Other Funding

• 2002-2005
  – US National Science Foundation grant to create a “gold standard” sub-corpus of 10 million words, hand-validated for logical structure, word and sentence boundaries, and POS
• December 2005: All funding sources expired
• Now seeking funding from US National Science Foundation, National Endowment for the Humanities, DARPA (Dept. of Defense), new consortium memberships

Availability

• Freely available to non-profit educational and research organizations from the outset
• No restrictions on obtaining the corpus based on geographical location
• Consortium members have exclusive access for commercial exploitation for 5 years
• Distributed by LDC
  – $75 distribution fee

Licensing

• LDC
  – Obtains licenses from text providers
  – Issues licenses to users
• Restricted portion
  – No redistribution without publisher’s permission
• Open portion
  – Licensed on the model of open-source software
  – Most 2nd release data under this license

Text Sources

• ANC consortium members
  – In fact have not contributed much data
• Corpus projects
  – Project MORE, ICSI, etc.
  – Mostly spoken
• Web
  – Requires validation that it is American English
  – Public domain or permission of copyright holder
• Contributions via ANC interface
  – Authors can upload texts, sign license, provide demographic info on the web
ANC Makeup

- Core “static” corpus
  - Texts and transcriptions of spoken data
  - 1990 onwards
  - Comparable in balance to BNC
    - Enables comparative studies
  - At least 100 million words
  - Snapshot of American English at the turn of the millenium

“Dynamic” component

- Not necessarily balanced
  - Dictated by availability
  - Includes email, ephemera, rap lyrics, blogs, newsgroups, etc.
  - Goal: add 10% every five years
  - Layered organization
    - Dynamic component layered chronologically as added

Eventual components?

- Annotated and aligned speech data
- Dialects of American and Canadian English
- Other major languages of North America
  - Spanish, French Canadian
  - aligned to parallel translations in English

High costs of production prevent inclusion at this stage

ANC 2nd Release

- First Release: October 2003
  - 11 million words
- Second Release: December 2005
  - 22 million words
  - Genres
    - Spoken (17%): telephone conversations, academic discourse, face-to-face conversations
    - Written (83%): technical articles, fiction, non-fiction, letters, journals, newspaper, government documents, travel guides, blogs

Annotations

- ANC 2nd Release is distributed with stand-off annotations for POS and lemma (2 different tagsets), noun and verb chunks
  - Possibly the first large scale, widely available corpus distributed in stand-off form
- Automatically produced, no validation
- Gold standard sub-corpus
  - 10 million words
  - Hand validated for token and sentence boundaries, part of speech (Biber tagset)
  - Proposal to add hand validated annotation for syntax, WordNet senses and FN frames, and named entities

Genre Distribution in 2nd release

- [Graph showing genre distribution]

American National Corpus
Derived Data

- We also freely distribute derived data for the ANC
  - Frequency lists
  - Bigrams, trigrams
    - Both word forms and POS
- Mark Davies (BYU) incorporating ANC data into his online interface for concordancing, etc.
  - Currently interfaces to the BNC
  - Available later this year
- University of Alberta provides an on-line concordance generator for ANC 1st release

Open Linguistic Infrastructure

- Adding annotations for a variety of linguistic phenomena
  - Use any software we can get our hands on
  - Freely available
  - Contributed
  - Some annotations already contributed
    - U Alberta: co-reference
    - Lancaster: CLAWS 5 and 7 POS tagging
    - Litkowski: Sense tags for nouns and adjectives
  - Multiple annotations of the same type
    - e.g., syntactic annotation: Charniak parser, Collins parser, Minipar, CMU Link parser...

Other Annotation Types

- WordNet sense tags
  - Several freely available or contributed sense taggers available
- Named entities
  - Persons, locations, organizations, dates, currency...
  - Others...
    - Time and events
- Frames
  - Frame elements (semantic roles, verb types)

OLI

- Assumption
  - Many unvalidated annotations of the same type can lead to ways to improve performance of automatic annotation software
    - Combine results
    - Machine learning
- Annotations freely available via the ANC website
  - Cannot freely distribute the corpus proper because of copyright issues
  - But annotations in stand-off form can be distributed freely
    - All in stand-off; anyone who has the corpus can use immediately

ISO, XCES, and ANC

- ISO TC37 SC4 has developed an overall model for annotations (Linguistic Annotation Framework)
- XCES is providing one possible instantiation format
- ANC is implementing it

Representation Model

- Annotations represented as a graph of feature structures
  - Nodes are locations in primary data or other annotations (“edge graph”)
  - Edges are labeled with feature structures containing the annotation content
- All ANC annotations transduced to this format
ANC Processing

- **Starting Point:**
  - documents in a wide variety of formats
    - Quark express, PDF (some multi-column), plain text, HTML, MS Word doc, RTF, SGML and XML...
- **Goal:**
  - render into XCES-compliant XML with stand-off annotations

The Process

- **Step 1**
  - May first require an underpaid student to turn Quark, PDF, etc. into something we can use
  - Use a suite of scripts in Perl, XSLT, etc. to turn the original format into XCES down to the paragraph level
  - NOTE: many creative uses of HTML, SGML, XML
    - HTML: focus on presentation (tables, tables, tables!), not always logical
    - SGML: use of `<p>` and `<quote>` tags to lay out tables
    - XML?: `<article>&lt;html&gt;&lt;head&gt;…</article>`

Step 2: GATE

- **GATE: General Application for Text Engineering**
  - Developed at University of Sheffield
  - Continually enhanced, supported
  - Framework to plug together all the tools we need
  - Freely distributed
  - In use by hundreds of researchers around the world
  - Note: we use GATE for our processing, no need for ANC users to use GATE (unless they want to)

Use of GATE

- **GATE built-in resources**
  - Tokeniser
  - POS tagger
  - Morphological analyzer
  - Named Entity recognition
  - Noun and verb chunker
  - Several modules for syntax
  - co-reference
- **ANC custom resources**
  - Sentence splitter (prevent sentence boundaries overlapping certain tags)
  - Biber POS tagger
  - Stand-off XML (XCES-ISO) markup generator
  - Resources to massage annotations and features

Problems

- Need to batch process large numbers of files
  - different problems in each
  - different types (spoken, written)
- Different resources required for different file sets
- Resource parameters depend on document type

Solution: XORO

- Scripting language for automating tasks in GATE, written by Keith Suderman (ANC programmer)
  - Includes syntactic sugar for using GATE resources and other Java classes
  - Batch process large groups of files in multiple directories
  - Flow of control allows for selectively applying different resources with different parameters to sets of documents
  - Allows rapid development of applications without recompiling Java code
  - Use from command line or as a plug-in to GATE
The Product

- ISO/LAF-compliant representation
- "Extreme standoff"
  - Original document is text only and read only
  - Each annotation in a separate XML document, linked to original
    - Allows for multiple (possibly overlapping) annotations, multiple annotations of the same type
    - Easy to modify and/or add to
    - Enables a distributed development model
    - Different sites independently add annotation
    - Suitable for delivery over the WWW
  - XML, RDF representation

LAF

- Under development by ISO Working Group of the sub-committee for Language Resources (ISO TC37 SC4)
- Basic concept is definition of an abstract ("pivot") format for annotations
  - Feature structures to represent annotation content
  - User can define any representation (XML, LALR, etc.) as long as it maps to the abstract format
  - Abstract format instantiated in XML
    - ANC uses the XML instantiation of the abstract format to represent annotations
    - Annotations are mapped into abstract format, then be translated to any other format

LAF Architecture

Life Cycle of an ANC Document

As many annotations as desired can reference the same segmentation or be layered over lower-level annotations
Annotations form a directed acyclic graph (DAG) over primary data

Advantages of DAG

- Can apply graph algorithms to traverse the graph
  - Breadth-first, depth-first traversal, shortest path, minimum spanning tree
  - Connectedness, articulation vertices
  - Topological sort
  - Graph coloring, graph partitioning
  - Etc.
- What can we do with this?
  - What is all info on path to/from node x
  - What is nearest common ancestor of nodes x and y
  - Find matching sub-graphs
  - Identify connected components
  - Which nodes (phenomena) are most connected, form articulation vertices, etc.

LAF Pivot Format

<?xml version="1.0" encoding="UTF-8"?
<cesAna xmlns="http://www.xces.org/schema/2003" version="1.0.4">
  <struct type="tok" from="27" to="28">
    <feat name="base" value="A"/>
    <feat name="msd" value="DT"/>
  </struct>
  <struct type="tok" from="29" to="34">
    <feat name="base" value="Brief"/>
    <feat name="msd" value="NNP"/>
  </struct>
  <struct type="tok" from="35" to="42">
    <feat name="base" value="History"/>
    <feat name="msd" value="NN"/>
  </struct>
  <struct type="tok" from="51" to="57">
    <feat name="base" value="Celtic"/>
    <feat name="msd" value="NNP"/>
  </struct>
  <struct>
    designates logical tree
    <struct> associates attributes with
      struct>’s can be nested to represent
      more complex structures
    Also tags to designate alternative
    conjunction
      A set of logical operations over
      structures is defined
  </struct>
</cesAna>

Problem

- Very little software that handles standoff annotations
  - Need a way to produce ANC data with annotations in-line
  - Need a way to merge annotations of the same or different types

XCES Parser

- A “SAX-like” parser that merges standoff annotations with the original text
  - Fires SAX events
  - Can be used with anything that uses a SAX parser, e.g. Saxon
- Produces a valid XML document
- Freely available on ANC web site

ANC Merge Tool

- Provides a graphical interface to the XCES parser for merging annotations and content
- Merges annotations of the user’s choosing with primary data into one document
- User chooses output format
- User chooses how to handle overlapping hierarchies
Output

- Output can be any format (not only XML)
  - We currently provide option to generate
    - MonoConc Pro format
    - Wordsmith
    - XML (usable in BNC’s XAIRA)
- Parser uses multiple implementations of the org.xml.sax.DocumentHandler interface, one for each output format
- Additional output formats easily generated by implementing additional interfaces

Merge Tool

- Schema-aware: can specify embeddings, priorities, things to include/omit, what is attribute/tag, etc.
- Can also be used by any application that allows the user to specify the SAX parser to be used
  - e.g., Saxon can be used to apply XSLT stylesheets to ANC data without first merging annotations and primary data

XML Format

```
<text>
  <body>
    <div type="article">
      <s>
        <head type="headline">
          <NounChunk>
            <tok base="harmonic" msd="NNP">Harmonic</tok>
            <tok msd="NNP" base="convergence" affix="s">Convergences</tok>
          </NounChunk>
        </head>
      </s>
      <p id="p1">
        <s id="p1s1">
          <NounChunk>
            <tok base="you" msd="PRP">You</tok>
          </NounChunk>
          <tok msd="VBP" affix="" base="be">'re</tok>
          <tok base="right" msd="NN">right</tok>,
          <NounChunk>
            <hi rend="ital">
              <tok base="maxim" msd="NNP">Maxim</tok>
            </hi>
            <tok base="'s" msd="POS">'s</tok>
            <NounChunk>
              <tok base="strong" msd="JJ">strong</tok>
              <tok base="point" msd="NN">point</tok>
            </NounChunk>
            <tok msd="VBZ" base="be" affix="s">is</tok>
            <tok base="that" msd="IN">that</tok>
            <NounChunk>
              <tok base="it" msd="PRP">it</tok>
            </NounChunk>
            <tok msd="VBZ" affix="s" base="be">'s</tok>
            <tok base="totally" msd="RB">totally</tok>
            <NounChunk>
              <tok base="unsentimental" msd="JJ">unsentimental</tok>
              <tok base="and" msd="CC">and</tok>
              <tok base="ungenteel" msd="NN">ungenteel</tok>
            </NounChunk>
            <tok base="." msd=".">.
          </s>
        </s>
      </p>
    </div>
  </body>
</text>
```

MonoConc Format

```
Harmonic_NNP Convergences_NNP

You_PRP're_VBP right_NN, Maxim_NNP 's_POS strong_JJ point_NN is_VBZ that_IN it_PRP's_VBZ totally_RB unsentimental_JJ and_CC ungenteel_NN.
```

Information

WEB: http://AmericanNationalCorpus.org
EMAIL: anc@cs.vassar.edu

ANC discussion list - join from ANC web page