

Grammatical annotation

L485/L700

Dept. of Linguistics, Indiana University
Autumn 2008

Desirable Linguistic Properties

We have established that, to analyze learner language, we need:

- ▶ Error annotation
 - ▶ to analyze errors
- ▶ General linguistic annotation
 - ▶ to compare/count correct usages vs. erroneous usages
 - ▶ to support error annotation
 - ▶ to train NLP models on correct usage

Today, we explore different types of linguistic, or grammatical, annotation and some general annotation principles

Leech's Seven Maxims of Annotation

1. It should be possible to remove the annotation from an annotated corpus in order to revert to the raw corpus.
2. It should be possible to extract the annotations by themselves from the text. This is the flip side of maxim 1. Taking points 1. and 2. together, the annotated corpus should allow the maximum flexibility for manipulation by the user.
3. The annotation scheme should be based on guidelines which are available to the end user.
4. It should be made clear how and by whom the annotation was carried out.

Leech's Seven Maxims (2)

5. The end user should be made aware that the corpus annotation is not infallible, but simply a potentially useful tool.
6. Annotation schemes should be based as far as possible on widely agreed and theory-neutral principles.
7. No annotation scheme has the a priori right to be considered as a standard. Standards emerge through practical consensus.

Annotation manuals are critical to understanding what the annotation is and what it means

- ▶ a list of annotations with brief explanations, i.e., list the tagset
 - ▶ NN1 singular common noun (e.g., *book*, *girl*)
- ▶ a specification of annotation practices
 - ▶ rules or guidelines for assigning annotation to specific text
 - ▶ often includes the “case law” of how to handle boundary phenomena

There will be boundary cases, and there will be arbitrary decisions which have to be made

- ▶ The documentation needs to clearly specify how to handle such cases
 - ▶ e.g., in the title *The War of the Worlds*, is *of* a proper noun?
- ▶ The importance principle is to maintain consistency across different texts and different annotators

The set of rules is necessarily incomplete, as new boundary cases continually emerge

How “good” an annotation scheme is depends upon two main notions:

- ▶ How desirable the annotation is for a given purpose: how linguistically plausible, cognitively real, descriptively adequate, etc.
- ▶ How accurately & consistently the annotation can be applied
 - ▶ Inter-rater agreement tests how much two annotators agree vs. how much they would agree by chance (kappa statistic)
 - ▶ Annotation error detection can highlight areas of inconsistency

Levels of Linguistic Annotation

Some of the possible levels of annotation:

- ▶ morphological annotation (e.g. inflection, derivation, compounding)
- ▶ morpho-syntactic annotation: part-of-speech (POS) tagging
- ▶ syntactic annotation (e.g. named entities, phrasal chunking, full syntactic analysis)
- ▶ semantic annotation (e.g. word-sense disambiguation, anaphora and coreference resolution, information structure)
- ▶ discourse annotation (e.g. dialog turns, speech acts)

Step by Step Annotation

Whether automatically or manually, these are common different layers of annotation, many of which are interrelated:

- ▶ tokenization
- ▶ lemmatization / morphological analysis
- ▶ part-of-speech tagging
- ▶ named-entity recognition
- ▶ partial parsing
- ▶ full syntactic parsing
- ▶ semantic and discourse processing

Lemmatization:

- ▶ refers to the process of relating individual word forms to their citation form (lemma) by means of morphological analysis
 - ▶ e.g. `stopped` \Rightarrow `stop`
- ▶ provides a means to distinguish between the total number of word tokens and distinct lemmata that occur in a corpus
 - ▶ e.g. helps to find all occurrences of `buy`
- ▶ is indispensable for highly inflectional languages which have a large number of distinct word forms for a given lemma

Lemmatization – German Example

Lemmas in second column

wie	wie	+Adv+Wh+#lex+COWIE
wie	wie	+Conj+Coord+#lex+COWIE
wie	wie	+Conj+Subord+#lex+COWIE
sie	sie	+Pron+Pers+3P+Pl+Fem+Nom+#lex+PERSPRO
sie	sie	+Pron+Pers+3P+Sg+Fem+Nom+#lex+PERSPRO
offenbar	offenbaren	+Verb+Imp+2P+Sg+#lex+VVFIN
offenbar	offenbar	+Adj+Pos+Pred+#lex+ADJD
gedacht	gedenken	+Verb+PPast+#lex+VVPP
gedacht	dachen	+Verb+PPast+#lex+VVPP
gedacht	denken	+Verb+PPast+#lex+VVPP
hat	haben	+Verb+Indc+3P+Sg+Pres+#lex+VAFIN

Morphological Analysis

Grammatical
annotation

L485/L700

Xerox:

half half+Adj

half half+Adv

half half+Noun+Sg

Corpus Annotation

Annotation principles

Linguistic annotation

Layers of
annotation

Lemmatization

POS Tagging

Parsing

Encoding

XML

Practical steps

Lingsoft:

"<half>"

"half" <Quant> DET PRE SG/PL @QN>

"half" <NonMod> <Quant> PRON SG/PL

"half" N NOM SG

"half" ADV

Part of Speech (POS) Tagging

POS Tagging = Assigning word class information to words

- ▶ In other words, disambiguating the morphological analysis in a given context

ex: *the* *man* *bought* *a* *book*
 determiner noun verb determiner noun

Questions:

- ▶ How do we divide the text into individual **word tokens**?
- ▶ How do we choose a **tagset** to represent all words?
- ▶ How do we select appropriate **tags** for individual **words**?

Tokenization has to deal with multiwords, merged words, and compounds

- ▶ Multiwords
 - ▶ e.g., *in spite of* the firm promise
- ▶ Merged words
 - ▶ he *couldn't* come, *dunno* == do not know
- ▶ Compounds
 - ▶ a *hundreds-of-billions-of-yen* market
 - ▶ *dieses Gestern-fand-ich-sie-noch-doof-heute-gehen-sie-schon-in-Ordnung-Feeling*

Possible Solution: Layered Analysis

```
<w pos=in> in spite of</w>
```

```
<w pos=md+rb> shouldn't</w>
```

```
<w pos=jj>  
  <w pos=nns> hundreds</w>  
  <w pos=in> of</w>  
  <w pos=nns> billions</w>  
  <w pos=in> of</w>  
  <w pos=nns> yen</w></w>  
<w pos=nn> market</w>
```

Issues in Selecting a Tagset

Grammatical
annotation
L485/L700

Corpus Annotation

Annotation principles
Linguistic annotation

Layers of
annotation

Lemmatization

POS Tagging

Parsing

Encoding

XML

Practical steps

- ▶ **conciseness**: short labels better than long ones

`prep` \Rightarrow `preposition`

- ▶ **perspicuity**: labels that are easily interpreted are better

`prep` \Rightarrow `in`

- ▶ **analysability**: should be possible to decompose in different parts

`vmfin`: **v**erb, **m**odal, **f**inite

`pds`: **p**ronoun, **d**emonstrative, **s**ubstituting

POS Representations

Grammatical
annotation

L485/L700

Horizontal Format

I/PP will/MD then/RB maybe/RB travel/VB
directly/RB on/IN to/IN Berlin/NP

Corpus Annotation

Annotation principles

Linguistic annotation

Layers of
annotation

Lemmatization

POS Tagging

Parsing

Encoding

XML

Practical steps

Vertical Format

I	PP
will	MD
then	RB
maybe	RB
travel	VB
directly	RB
on	IN
to	IN
Berlin	NP

Annotating POS Tags

Two main, different approaches:

- ▶ start from scratch, find characteristics in words or context (= rules) which give indication of word class
i.e. if word ends in ``ion'', tag it as noun
- ▶ accumulate lexicon, disambiguate words with more than one tag
i.e. possible categories for ``about``:
preposition, adverb, particle

Elements of both can be used in developing manual annotation

- ▶ Often: automatically tag and then post-edit

Some available POS Taggers

- ▶ Amalgam tagger - Email: mail-in tagger platform, supports different corpora and tagsets for English
link: <http://www.comp.leeds.ac.uk/amalgam/amalgam/amalgtag3.html>
- ▶ Brill tagger: transformation-based error driven learning
link: <http://research.microsoft.com/users/brill/>
- ▶ TnT (Tags and Trigrams): Hidden Markov Model, best tagger available
link: <http://www.coli.uni-saarland.de/~thorsten/tnt/>
- ▶ TreeTagger: decision tree tagger, has pre-trained models for a variety of languages
link: <http://www.ims.uni-stuttgart.de/projekte/complex/TreeTagger/DecisionTreeTagger.html>

(1) Analyzing a sentence into its constituents, identifying in greater or less detail the syntactic relations and parts of speech.

(2) Describing a word in a sentence, identifying its part of speech, inflectional form, and syntactic function.

McArthur (1992) The Oxford Companion to the English Language.

These issues of course become more problematic when learner data is being analyzed

- ▶ partial (chunk) parsing vs. full parsing
 - ▶ efficiency/robustness vs. amount of information
- ▶ “normal” context-free grammar (CFG) parsing vs. parsers based on more complex grammar formalisms
 - ▶ Are all the relevant properties being captured?
- ▶ all parses vs. most probable one
 - ▶ Do we want to retain ambiguity?
- ▶ write grammar manually vs. parser trained on treebank
 - ▶ development time, reusability, accuracy, ...

Partial Parsing

- ▶ parsing “islands of certainty” first
- ▶ partial analysis: parsing only phrase boundaries and clause boundaries
- ▶ no explicit attachment of phrases
- ▶ generally no grammatical functions

Chunks

- ▶ chunk: non-recursive kernel of a phrase
kernel: no right-modification

- ▶ why?

on Sunday

on Sunday after the Easter weekend

a \$ 1 billion budget deficit

The soft-spoken, silver-haired Manhattan borough president

the song “I need a woman tonight”, which caused major problems

union leaders and political cronies who may seek a place at the trough

a convicted kidnapper who later said publicly that he considers himself anti-white

a Manhattan city councilwoman, some of whose programs, such as commercial rent control, have made their way into Mr. Dinkins 's position papers

Overgeneration

- ▶ goal of linguists in grammar writing: cover *all grammatical* sentences of a language but *no ungrammatical* ones
- ▶ problem: rules follow *Zipf's law*: few rules occur very often, many rules occur only once
- ▶ in general: if you add one sentence to a treebank, you also add one new rule
- ▶ grammar never finished?
- ▶ other approach: describe *as many grammatical structures* as possible, don't worry about ungrammatical ones \Rightarrow overgeneration
- ▶ overgeneration makes parser **robust**

Some available Chunk Parsers

- ▶ CASS: Steven Abney's chunk parser
extremely fast, purely deterministic bottom-up chunk parsing
URL: <http://www.vinartus.net/spa/>

- ▶ TTT: text tokenization system and toolset, University of Edinburgh, Language Technology group
more powerful, uses XML as input/output format
URL: <http://www.ltg.ed.ac.uk/software/ttt/>

Grammatical
annotation
L485/L700

Corpus Annotation

Annotation principles
Linguistic annotation

Layers of
annotation

Lemmatization
POS Tagging

Parsing

Encoding

XML

Practical steps

Some available Full Parsers

- ▶ **LoPar**: probabilistic chart parser, can be lexicalized implemented by Helmut Schmid, Stuttgart
URL: <http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html>
- ▶ **Stanford Parser**: Java implementation of probabilistic natural language parsers, both highly optimized PCFG and dependency parsers, and a lexicalized PCFG parser implemented by Dan Klein, Chris Manning et al.
URL: <http://nlp.stanford.edu/downloads/lex-parser.shtml>
- ▶ **Charniak parsers**: most recent parsers, optimized for Penn Treebank implemented by Eugene Charniak, Brown University
URL: <http://www.cs.brown.edu/people/ec/#software>

Text Encoding Initiative (TEI)

- ▶ project sponsored by the Association for Computational Linguistics, the Association for Literary and Linguistic Computing, and the Association for Computers in the Humanities
- ▶ encoding guidelines
- ▶ link: <http://www.tei-c.org>
- ▶ define how documents should be marked-up with the mark-up language SGML (or more recently XML)

- ▶ XML: Extensible Markup Language
 - ▶ Used to structure data
 - ▶ similar to HTML
- ▶ has no fixed “semantics”: user defines what tags mean
 - ▶ formally verifiable via document type definitions (DTD) of XML Schema Definitions (XSD)
- ▶ recognized as international ISO standard
- ▶ tools available for editing, displaying, querying

XML – Example

```
<CATALOG>
  <CD>
    <TITLE>Empire Burlesque</TITLE>
    <ARTIST>Bob Dylan</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>Columbia</COMPANY>
    <PRICE>10.90</PRICE>
    <YEAR>1985</YEAR>
  </CD>
  <CD>
    <TITLE>Greatest Hits</TITLE>
    <ARTIST>Dolly Parton</ARTIST>
    <COUNTRY>USA</COUNTRY>
    <COMPANY>RCA</COMPANY>
    <PRICE>9.90</PRICE>
    <YEAR>1982</YEAR>
  </CD>
</CATALOG>
```

Corpus Annotation

Annotation principles
Linguistic annotation

Layers of annotation

Lemmatization
POS Tagging
Parsing

Encoding

XML

Practical steps

Each text that conforms with the TEI guidelines consists of two parts— a header and the text itself.

The header contains information such as:

- ▶ author, title, and date
- ▶ the edition or publisher used in creating the machine-readable text
- ▶ information about the encoding practices adopted

XML Annotated Text

Grammatical
annotation

L485/L700

Corpus Annotation

Annotation principles
Linguistic annotation

Layers of
annotation

Lemmatization
POS Tagging
Parsing

Encoding

XML

Practical steps

```
<text>
<body>
<div type=BODY>
<div type="Q">
<head>Subject: The staffing in the Commission of the European
Communities
</head>
<p>Can the Commission say:</p>
<p>1. how many temporary officials are working at the Commission?</p>
<p>2. who they are and what criteria were used in selecting them?</p>
</div>
<div type="R">
<head>Answer given by <name type=PERSON><abbr rend=TAIL-SUPER>
Mr</ABBR> Cardoso e Cunha</name> on behalf of the Commission
<date>(22 September 1992)</date></head>
<p>1 and 2. The Commission will send tables showing the number of
temporary staff working for the Commission directly to the
Honourable Member and to Parliament's Secretariat.</p>
</div></div></body></text>
```

Doing (manual) annotation

- ▶ Acquire the necessary data (which has issues of representativeness, etc.)
- ▶ Annotate the text, perhaps some combination of automatic tools and hand-tagging (post-editing)
 - ▶ This can be done with a simple text editor
 - ▶ If you want XML, you can do column format and then run a program to convert it to XML later
 - ▶ You can also use an annotation tool like MMAX2
- ▶ Validate the annotation
 - ▶ XML schemas enforce XML well-formedness
 - ▶ Running multiple annotators over certain parts of the corpus can identify problem spots
 - ▶ Annotation error detection tools also detect problematic cases
- ▶ Revise the guidelines as you go