

Building and searching large parsed corpora of diachronic texts

Beatrice Santorini

University of Pennsylvania

<http://www.ling.upenn.edu/~beatrice/corpus-ling.html>

Deutsch Diachron Digital

10 December 2003

Goals and principles of annotation

- Corpus consists of straight-up ASCII
 - Syntactic annotation is represented as labeled bracketing
 - No internal formatting codes
 - No dependence on obsolescent software
- Annotated corpus = God's truth, **not**
 - The primary goal of our annotation is to facilitate searches for various constructions of interest.
 - The goal is **not (!)** to associate every sentence with a correct structural description.

Dealing with uncertainty and ambiguity

- As many syntactic categories as possible should have clear meanings so that the number of unclear cases is minimized.
- We try to avoid controversial decisions.
- To that end, we sometimes omit information.
 - VP boundaries
 - Subtle distinctions (adjectival vs. verbal passives, argument vs. adjunct PPs)
- In other cases, we use default rules.
 - Location of wh-traces
 - PP attachment (“when in doubt, attach high”)

An example of diachronic ambiguity

- ((CP-QUE (WNP-1 Which house)
 (IP-SUB (MD did)
 (NP-SBJ they)
 (NP-OB1 *T*-1)
 (VB buy)
 (. ?)))
- ((CP-QUE (WNP-1 Which house)
 (IP-SUB (MD did)
 (NP-SBJ they)
 (VB buy)
 (NP-OB1 *T*-1)
 (. ?)))

An incorrect, yet useful, structure

Our solution: we consistently put the trace in a position that is **linguistically unmotivated**.

```
( (CP-QUE (WNP-1 Which house)
  (IP-SUB (NP-OB1 *T*-1)
    (MD did)
    (NP-SBJ they)
    (VB buy)
  (. ?))))
```

Example sentence 1

```
( (IP-MAT (NP-SBJ=1 (EX There))
  (BEP is)
  (NP-1 (ONE one) (NPR M=r=) (NPR Colson)
    (CP-REL (WNP-2 0)
      (C 0)
      (IP-SUB (NP-SBJ (PRO I))
        (BEP am)
        (ADJP (ADJ shure)
          (CP-THT (C 0)
            (IP-SUB (NP-ACC *T*-2)
              (NP-SBJ (PRO$ my) (N Lady))
              (HVP has)
              (VBN seen)
              (PP (P at)
                (NP (N diner)
                  (PP (P w=th=)
                    (NP (PRO$ my)
                      (N Unckle)))))))
          )
        )
      )
    )
  )
  (. .)) (ID ALHATTON,2,241.7))
```

Example sentence 2

```
( (IP-MAT (NP-SBJ (PRO I))
  (HVP have)
  (NP-ACC (NP-MSR (QP (ADVR so) (Q much)
    (CP-DEG *ICH*-1))))
  (N buisness)
  (ADVP-LOC (ADV here))
  (CP-DEG-1 (C y=t=)
    (IP-SUB (NP-SBJ (PRO I))
      (VBP hope)
      (CP-THT (C 0)
        (IP-SUB (NP-SBJ (PRO$ my) (N Lady))
          (MD will)
          (VB excuse)
          (NP-ACC (PRO me))
          (PP (P till)
            (NP (ADJS next)
              (N post))))))))))
(. .)) (ID ALHATTON,2,245.46))
```

<----- not a typo!

How we build a parsed corpus - a flowchart

- POS tagging
 - Automatic preprocessing (punctuation, contractions)
 - Automatic tagging (Brill 1995)
 - Human correction
- Parsing
 - Automatic parsing (Collins 1996)
 - Human editing (= correction + addition of information)
- Final editing (partially automated)

Correction software

- We use correction software developed in connection with the Penn Treebank (<http://www.cis.upenn.edu/~treebank>) and implemented in Emacs Lisp
- Incorrect tags are corrected by positioning cursor on item to be corrected and entering correct tag
- Proposed tag is checked to ensure that new tag is legal
- Correction software leaves input text inviolate

Project management

- Mean editing speed (in language well-known to annotator):
2,000 words/hours for POS-tagging
1,000 words/hours for parsing
- Annotators can work approx. 4 hours/day or 20 hours/week
- Annotators are relatively easy to find and train for POS-tagging, but quite a bit harder to find and train for parsing (people are used to thinking about words, but not in terms of constituent structure)

So how long does it take to produce a parsed corpus of 1 M words?

- POS-tagging stage
 - $1,000,000 \text{ words} / 2,000 \text{ words/hours} = 500 \text{ hours}$
 - $500 \text{ hours} / 20 \text{ hours/week} = 25 \text{ weeks}$
- Parsing stage
 - $1,000,000 \text{ words} / 1,000 \text{ words/hours} = 1,000 \text{ hours}$
 - $1,000 \text{ hours} / 20 \text{ hours/week} = 50 \text{ weeks}$
- Total: 75 weeks

A search engine for parsed corpora

- A corpus without a search program is like the Internet without Google.
- Enter CorpusSearch (Randall 2000), a dedicated search engine for parsed corpora
- Written in Java
- Runs under Linux, Mac, Windows, Unix

A key feature: Searchable output

- Complicated and error-prone monster queries can be implemented as a sequence of simpler queries.
- Sequences of queries are consistent with the way that corpus research proceeds, via a successive refinement of hypotheses.
- Generating searchable output slows CorpusSearch down somewhat (searches of 1-2M words can take 2-3 minutes)

Basic search functions are linguistically intuitive

- exists
- precedes
- immediately precedes
- immediately dominates

Simple **dominate** discontinued in CorpusSearch 1.1, but easy to simulate

A simple sample query

node: IP*

query: (IP* iDoms NEG)

- Asterisk is a wildcard
(IP* matches IP-MAT, IP-SUB, IP-INF, etc.)
- CorpusSearch searches the corpus for constituents with the label(s) specified in **node**.
- Whenever it finds such a constituent, it checks whether the material in the constituent matches the condition(s) in **query**.
- Matching tokens are recorded in an output file.

Definition files

- Let's say we want to find NP objects in the Vorfeld:
 - Den Lothar habe ich recht gern.
 - Dem Lothar ist nicht zu trauen.
 - Des Lothars gedenken wir selten.
- We don't want other constituents in the Vorfeld:
 - Nächste Woche treffe ich den Lothar.
 - Mit dem Lothar verstehe ich mich gut.
 - Nur selten gedenken wir Lothars.

A possible query, but long-winded and error-prone

node: S*

```
query: ((S* iDomsNum1 NP-AKK | NP-DAT | NP-GEN)
        AND
        (S* iDomsNum2 SEIN-PRS | SEIN-PRT |
          HAB-PRS | HAB-PRT |
          MODAL-PRS | MODAL-PRT |
          VERB-PRS | VERB-PRT))
```

A better way

define: v2.def

node: S*

query: ((S* iDomsNum1 Objekt)
AND
(S* iDomsNum2 Vb-fin))

Contents of the definition file v2.def:

Objekt: NP-AKK | NP-DAT | NP-GEN

Vb-fin: SEIN-PRS | HAB-PRS | MODAL-PRS | VERB-PRS |
SEIN-PRT | HAB-PRT | MODAL-PRT | VERB-PRT

Using definition files to construct word classes

- Definition file defines:

```
verb:      VB | VBP | VBD | VAG | VAN
```

```
drink:     drink | drinks | drank | drunk | drinking
```

```
eat:       eat | eats | ate | eaten | eating
```

```
read:      read | reads | reading
```

```
write:     write | writes | wrote | written | writing
```

```
TR-INTR:  \ $drink | \ $eat | \ $read | \ $write
```

- Query file makes reference to word class:

```
query: (verb iDoms TR-INTR)
```

Ignoring material in searching for V3

- We certainly want this sentence to count as V3:
 - (PP In Jahre 1356,) (NP-SBJ der Kaiser) besuchte den Papst.
- But we might want these sentences to count as V2:
 - (CONJ Und) (NP-AKK den Lothar) treffe ich morgen.
 - (INTJP Ojemine,) (NP-AKK den Lothar) treffe ich wohl nicht mehr.
 - (NP-LFD-1 Den Lothar,) (NP-AKK=1 den) treffe ich morgen.
 - (NP-VOC Du,) (NP-AKK den Lothar) treffe ich erst im Januar.

The V3 query

definition: v2.def

node: S*

add_to_ignore: CONJ | INTJP | NP-LFD* | NP-VOC

query: (S* iDomsNum3 Vb-fin)

The asterisk after NP-LFD is necessary because left-dislocated constituents are coindexed with resumptive expressions,

A cool feature: The coding function

- Ordinary queries generate one output file per query.
- This can lead to an unwieldy proliferation of output files.
- Moreover, we are often interested in multivariate statistical analysis.

An example from this history of English

- Verb raising (old grammar):
He loves me, he loves me not.
- *Do* support (new grammar):
She loves me, she does not love me.

A coding query

```
1: { o: (finite_verb precedes NEG)
     n: ((finite_do precedes NEG) AND (NEG precedes VB)) }

2: { 1: (*AMBASS* inID)
     3: (*ANHATTON* inID)
     1: (*APOOLE* inID)
     2: (*ARMIN* inID)
     3: (*AUNGIER* inID)
     2: (*AUTHNEW* inID)
     2: (*AUTHOLD* inID) }

3: { f: (*ANHATTON* inID)
     f: (*APOOLE* inID)
     m: ELSE }

4: { v: (*ANHATTON* inID)
     v: (*APOOLE* inID)
     v: (*ARMIN* inID)
     b: (*AUTHNEW* inID)
     b: (*AUTHOLD* inID)
     n: ELSE }
```


- See handout for output of coding query.
- Coding strings can include blank slots for subsequent manual analysis. This is useful in analyzing discourse factors.
- Coding strings are compatible with standard programs for multivariate analysis of linguistic variation.

- Output of coding queries can serve as input to subsequent coding queries. This is useful for analyzing statistical interaction.

For purposes of illustration, the following query groups nonvernacular texts written by women as distinct from all other combinations of genre and sex.

```
5: {  
    a: ((CODING col4 n) AND (CODING col3 f))  
    b: ELSE  
}
```

Developments in the works - CorpusSearch

- Better negation
- Better disjunction
- Extended support for regular expressions
- “Search and replace” annotation support

Extended support for regular expressions

- CorpusSearch 1.1 requires:

```
kein:      kein | Kein | keyn | Keyn
keine:     keine | Keine | keyne | Keyne
keinem:    keinem | Keinem | keynem | Keynem
keinen:    keinen | Keinen | keynen | Keynen
keiner:    keiner | Keiner | keyner | Keyner
keines:    keines | Keines | keynes | Keynes
KEIN:      $kein | $keine | $keinem | $keinen |
           $keiner | $keines
```

- CorpusSearch 2 will allow:

```
kein:      [kK]e[iy]ine*[mnrs]*
```

Developments in the works - Parsing

- A parser being developed at Penn (Bikel, dissertation in progress) lets the user modify linguistic parameters, allowing increased crosslinguistic flexibility.
- The same parser allows multiple passes through a corpus, each pass respecting the previous ones.
- Advantages of multiple pass syntactic annotation
 - Multiple passes simplify the editing task (“divide et impera”)
 - Simplification means improvements in speed and consistency
 - Editing could be carried out by a mixture of more and less highly trained annotators.

((IP-MAT (NP-SBJ *pro*)
 (VBP thank)
 (NP-OB2 (PRO you))
 (PP (P for)
 (NP (PRO\$ your)
 (N attention))))
 (. .)))

((NP (Q Any) (NS questions))
 (. ?))