What a parsed corpus is and how to use it

Anthony Kroch and Beatrice Santorini
University of Pennsylvania

LSA Summer Institute Workshop on Diachronic Syntax

June 29-30, 2013
Types of annotation

- **Lemmatization**
  Word sense disambiguation
  Spelling normalization

- **Morphological tagging**
  Case, gender, number features on nouns
  Tense, mood, aspect features on verbs

- **Part-of-speech (POS) tagging**
  Elementary syntactic functions

- **Syntactic parsing**
  Hierarchical structure of phrases / clauses
  Grammatical function of phrases / clauses
POS tags

• POS tags contain elementary syntactic information

• They may also contain some morphological information

• More morphological information for some stages / languages than for others
A sentence with POS tags

( (PRO They) 
 (HVP have) 
 (D a) 
 (ADJ native) 
 (N justice) 
 (, ,) 
 (WPRO which) 
 (VBP knows) 
 (Q no) 
 (N fraud) 
 (. ;))

Syntactic tags

• Grammatical functions are indicated by dash tags, not configurationally

• Various difficult decisions are avoided
  No distinction between PP arguments and adjuncts
  No VP (more on this later)

• Not all grammatical functions are indicated
  No dash tags for PPs
The sentence with syntactic tags

( (IP-MAT (NP-SBJ (PRO They)))
  (HVP have)
  (NP-OBI (D a)
    (ADJ native)
    (N justice)
    (,)
  )
  (CP-REL (WNP (WPRO which))
    (IP-SUB (VBP knows)
      (NP-OBI (Q no)
        (N fraud))))
  )
)
Keeping it simple

- Some corpora use standoff annotation (text and annotation belong to different files)
- In the corpora discussed here, the text and the annotation belong to the same file
  
  Simpler corpus construction
  
  Simpler searches
  
  Simpler revision
  
  Simpler software for all of the above
Other syntactic information

- Traces indicate wh-movement
- Other empty categories, including empty complementizer, various types of empty subject
- Verb movement not indicated
- Also added to each token:
  - Text source and other philological information
The sentence, final version

( (IP-MAT (NP-SBJ (PRO They)))
 (HVP have)
 (NP-OBI (D a))
 (ADJ native)
 (N justice)
 (, ,)
 (CP-REL (WNP-1 (WPRO which)))
 (C 0)
 (IP-SUB (NP-SBJ *T* -1))
 (VBP knows)
 (NP-OBI (Q no)
 (N fraud))))))
 (. ;))
 (ID BEHN-E3-P1,150.48))
What is the purpose of an annotated corpus?

• Not (!) intended to represent God's truth

  Certainly impossible for languages undergoing change

  Impossible even for one that are grammatically stable
• God's truth is elusive

  Be that as it may, even given these problems, we decided a long time ago to forge ahead, come what might.

• Theoretical assumptions change, as do notations

• Context doesn’t always resolve semantic ambiguity

• Structural ambiguity is pervasive
Ambiguity during change

• OV > VO
  Wh- traces preverbal or postverbal?
  OV surface order basic or due to leftward movement?
  Mutatis mutandis for VO surface order

• V2 > non-V2
  SVO surface order V2 or not?
Attachment ambiguity

- They fight never.
- They will never fight. (85%)
  They never will fight. (15%)
- They never fight.
- They ___ never fight.
  They never ___ fight.
Dealing with ambiguity

- Omit some structure
  No verb movement
  No VP

- Establish default rules
  Wh- traces are clause-initial
  If in doubt, attach high
  Indirect question trumps free relative
What is the purpose of an annotated corpus?

The purpose is to facilitate the retrieval of sentences with particular linguistic properties of interest.
Searching a corpus

A corpus without a search program is like the Internet without a search engine (Beth Randall)
Diagnostic sentence types for loss of V2

- V2
  \[ \text{XP} \gg V\text{-fin} > \text{Sbj} \]

- non-V2
  \[ \text{XP} \gg \text{Sbj} > V\text{-fin} \]
V2 sentence

( (IP-MAT (PP (P In)
  (NP D +tat) (N book)))
  (BED were)
  (NP-SBJ (D +te)
    (VAN forsayd)
    (NS lawes)))
  (VAN y-write)
  (. ;))
  (ID CMPOLYCH-M3, VI, 35.229))
Non-V2 sentence

( (IP-MAT (CONJ And)
  (ADVP-TMP (ADV +tan))
  (NP-SBJ (D the) (N fuyre))
  (VBD cese.de)
  (.,))
  (ID CMPOLYCH-M3,VI,13,81))
Using definitions files

Sbj: NP-NOM* | NP-SBJ*

XP: ADVP* | NP-OB1* | NP-OB2* | PP*

V-fin: BED | BEP | DOD | DOP | HVD | HVP |
        MD | VBD | VBP

alternatively:

Query for V2 sentences

query:  (IP-MAT* iDomsNum 1 XP)
   AND (IP-MAT* iDomsNum 2 V-fin)
   AND (IP-MAT* iDoms Sbj)
   AND (IP-MAT* domsTotal< 10)
Query for non-V2 sentences

query: (IP-MAT* iDomsNum 1 XP)
AND (IP-MAT* iDomsNum 2 Sbj)
AND (IP-MAT* iDoms V-fin)
AND (IP-MAT* domsTotal< 10)
Wait a minute...

- The non-V2 sentence and the non-V2 query don’t match up!
- The first immediate constituent of the non-V2 sentence is CONJ
- The first immediate constituent in the query is XP
- XP doesn’t include CONJ
- So how did the query retrieve the sentence?
Ignoring syntactic labels

- Punctuation
- Conjunctions
- Interjections
- Vocatives
- Parentheticals
- Left-dislocates constituents
- Clitics
Query types

- Ordinary queries
- Coding queries
- Revision queries
Coding queries

• Ordinary queries search a corpus and report the matching sentence tokens in a separate output file

• Each query corresponds to a particular sentence type

• Coding queries allow information to be recorded that results from many separate ordinary queries

• The information is added to each sentence token in the form of coding strings
Sample coding query output

( (IP-MAT (CODING advp : pro : sbj-v : dirV))
  (ADVP (ADV Here))
  (NP-SBJ (PRO we))
  (VBP go)))

( (IP-MAT (CODING pp : np : v-sbj : dirV))
  (PP (P Around))
    (NP (D the) (N corner)))
  (VBD came)
  (NP-SBJ (D the) (N bus)))
Coding query for column 1

1: {  
  subj: (IP-MAT* iDomsNum 1 NP-SBJ*)  
  dir:  (IP-MAT* iDomsNum 1 NP-OB1*)  
  ...
  advp: (IP-MAT* iDomsNum 1 ADVP*)  
  pp:  (IP-MAT* iDomsNum 1 PP*)  
  ...
  -: ELSE

}
Coding query for column 2

2: { conj: (IP-MAT* iDoms NP-SBJ*)

AND (NP-SBJ* iDoms CONJP)

pro: (IP-MAT* iDoms NP-SBJ*)

AND (NP-SBJ* iDomsOnly PRO)

...

np: (IP-MAT* iDoms NP-SBJ*)

-: ELSE

}


Coding query for column 3

3: {  
sbj-v:  (IP-MAT* iDoms NP-SBJ*)
    AND (NP-SBJ* hasSister V-fin)
    AND (NP-SBJ* precedes V-fin)
  
v-sbj:  (IP-MAT* iDoms NP-SBJ*)
    AND (NP-SBJ* hasSister V-fin)
    AND (V-fin precedes NP-SBJ*)
  
:-: ELSE

}
Coding query for column 4

4: { dirV: (IP-MAT* iDoms V*)
    AND (V* iDoms go | went | gone |
        ... | come | came | ...
    )
    ordV: (IP-MAT* iDoms V*)
    :- ELSE
}

Poor man’s lemmatizer

come:    [cC][ao]me | [cC]omes | [cC]ometh |
         [cC]omeing* | [cC]om[iy]ng* | ...
go:     [gG]o | [gG]one | [gG]oes | [gG]oeth |
         [gG]o[iy]ng* | [gG]on* | [gG]oon* | ...
         [wW]ent* | [wW]hent* | ... | [eE]od* | ...
Coding query for column 4, revised

4: { dirV: (IP-MAT* iDoms V*)
    AND (V* iDoms $go | $come)
    ...
    ordV: (IP-MAT* iDoms V*)
    -: ELSE
}

How do the coding strings get used?

• The coding strings alone can be written to a file
  
  `advp : pro : sbj-v : dirV`
  `pp : np : v-sbj : dirV`
  `dir : pro : sbj-v : ordV`
  ...

• The file can then be exported for analysis to standard statistical software
Why revision queries?

In the analysis of V2 in the history of English, we want to track the following sentence schemas

\[
\begin{align*}
XP & \quad \text{Sbj-NP} \quad \text{V-fin} \quad \ldots \\
XP & \quad \text{Sbj-pro} \quad \text{V-fin} \quad \ldots \\
XP & \quad \text{V-fin} \quad \text{Sbj-NP} \quad \ldots \\
XP & \quad \text{V-fin} \quad \text{Sbj-pro} \quad \ldots 
\end{align*}
\]
Diagnostic sentence types for V2 in Old English

V2

AdvP  V-fin  Sbj-NP ...
AdvP Sbj-pro  V-fin ...
AdvP Sbj-pro Obj-pro Obj-pro V-fin ...

Non-V2

PP  Sbj-NP  V-fin ...
Problem, cont’d

• We want to ignore object pronouns
• We don’t want to ignore subject pronouns
• So we can’t just add PRO to the ignore list
Solution: Revision queries

- Revision queries allow users to add information to (a copy of) the corpus
- In contrast to coding queries, revision queries don’t just add coding strings
- Rather, they modify the actual annotation
Sample revision query

query: \[(IP-MAT^* \text{idoms} \{1\}NP-OB1^* \mid NP-OB2^*)\]

AND \[(NP-OB1^* \mid NP-OB2^* \text{idomsOnly PRO})\]

prepend_label \{1\}: IGNORE-
Sample revision query output

( (IP-MAT (PP (P on)

(NP (D +t+an)

(ADJ +triddan)

(N mon+de))

(IGNORE-NP-OB1 (PRO hiene)

(NP-SBJ (PRO man))

(RP+VBD ofslog)

(., .))

(ID coorosiu, Or_6:23.144.18.3029))
Ordinary V2 query, revised

add_to_ignore: IGNORE-*

query:      (IP-MAT* iDomsNum 1 XP)
            AND (IP-MAT* iDomsNum 2 V-fin)
            AND (IP-MAT* iDoms Sbj)
More on revision queries

• Revision queries can greatly simplify complex searches or even make them possible at all

• Queries containing many common search properties can be simplified and speeded up by “predigesting” the corpus to factor out the common properties

• Corpora of various origins can be made to conform to a single set of annotation conventions
Yet more on revision queries

• Revision queries greatly speed up corpus correction, especially when run in suites

• They can be used to construct training corpora for parsers

• In fact, we have used revision queries instead of standard parsers to build entire corpora
The end

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

(ID LSA-2013-06-28,42))