Using the PennParsed Corpora of Historical English with CorpusSearch

Waseda Workshop on the PPCHE

Anthony Kroch & Beatrice Santorini
University of Pennsylvania
December 12, 2017
• Slides for this workshop
  
  www.ling.upenn.edu/~kroch/handouts/

• CorpusSearch user’s guide
  
  corpussearch.sourceforge.net/CS-manual/Contents.html

• Annotation manual for PPCHE
  
  www.ling.upenn.edu/~beatrice/annotation/
What is a morphosyntactically annotated corpus?
• **morphological tagging**
  case, gender, number features on nouns
tense, mood, aspect features on verbs, etc.

• **lemmatization**
  word sense disambiguation
  spelling normalization

• **part of speech tagging**
  elementary syntactic functions

• **syntactic parsing**
  hierarchical structure of phrases/clauses
  grammatical function of phrases/clauses
An example sentence

(((IP-MAT (NP-SBJ (PRO They)))
  (HVP have)
  (NP-OBJ1 (D a)
    (ADJ native)
    (N justice)
    (, ,)
  )
  (CP-REL (WNP-1 (WPRO which)))
  (C 0)
  (IP-SUB (NP-SBJ *T*-1)
    (VBP knows)
    (NP-OBJ1 (Q no)
      (N fraud)))))))

(.,)
(ID BEHN-E3-PI,150.48))
They have a native justice, which knows no fraud;  (BEHN−E3−P1,150.48)
Reformated sentence for visualization

(((IP-MAT (NP-SBJ (PRO They)) (HVP have))
  (NP-OB1 (D a) (ADJ native) (N justice) (, ,))
  (CP-REL (WNP-1 (WPRO which)) (C 0))
  (IP-SUB (NP-SBJ (*T*-1) (VBP knows))
    (NP-OB1 (Q no) (N fraud))))))
  (. ;))
  (ID BEHN-E3-P1,150.48))
The example with lemmatization

(((IP-MAT (NP-SBJ (PRO (ORTHO They) (METAWORD (LEMMA (HEADWORD they) (OEDID 200700))))))

(HVP (ORTHO have) (METAWORD (LEMMA (HEADWORD have) (OEDID 84705))))

(NP-OB1 (D (ORTHO a) (METAWORD (LEMMA (HEADWORD a) (OEDID 4))))

(ADJ (ORTHO native) (METAWORD (LEMMA (HEADWORD native) (OEDID 125304))))

(N (ORTHO justice) (METAWORD (LEMMA (HEADWORD justice) (OEDID 102198))))

(, (ORTHO ,) (METAWORD (LEMMA (HEADWORD ,) (OEDID NA)))))
(CP-REL (WNP-1 (WPRO (ORTHO which)
  (METAWORD
    (LEMMa (HEADWORD which)
      (OEDID 228284))))
  (C (METAWORD (ALT-ORTHO 0)
        (LEMMa 0)))))

(IP-SUB (NP-SBJ (METAWORD (ALT-ORTHO *T*-1)
                   (LEMMa 0)))
         (VBP (ORTHO knows)
            (METAWORD
              (LEMMa (HEADWORD know)
                (OEDID 104157))))
         (NP-ACC (Q (ORTHO no)
                    (METAWORD
                      (LEMMa (HEADWORD no)
                        (OEDID 127437))))
                    (N (ORTHO fraud)
                        (METAWORD
                          (LEMMa (HEADWORD fraud)
                            (OEDID 74298))))))
         (.) (ORTHO ;)
         (METAWORD
          (LEMMa (HEADWORD .)
            (OEDID NA))))
         (ID BEHN-E3-P1,150.48))
• morphological tagging
case, gender, number features on nouns
tense, mood, aspect features on verbs, etc.

• lemmatization
word sense disambiguation
spelling normalization

• part of speech tagging
elementary syntactic functions

• syntactic parsing
hierarchical structure of phrases/clauses
grammatical function of phrases/clauses
The annotation task

- Annotation is multilevel and complex, so that using human effort for the whole job is impractical.
- At the same time, accuracy is crucial and unattainable at present with fully automated methods.
- In consequence, parsed corpora are built by interleaving automated analysis with human correction of the output.
Annotation software

- Wide range of software for automatic part-of-speech tagging and other software for automatic parsing.
- Software for correcting the errors of automated taggers.
- **Annotald** software for the correction of the errors of automatic parsers (annotald.github.io).
- **CorpusSearch** revision queries for semi-automatic parsing and parsing correction.
Available parsed corpus resources for European languages using the Penn annotation scheme
English Parsed Corpora, I


  1.3 million words


  1.8 million words


  3.0 million words
English Parsed Corpora, II


A sample of other languages, I

• Eiríkur Rögnvaldsson et al. *Icelandic Parsed Historical Corpus (IcePaHC)*, version 0.9, 8/2011. ([http://linguist.is/icelandic_treebank/Icelandic_Parsed_Historical_Corpus_(IcePaHC)](http://linguist.is/icelandic_treebank/Icelandic_Parsed_Historical_Corpus_(IcePaHC)))

  ≈1 million words


  ≈1 million words


  ≈2 million words, 0.8 million parsed to date
Other languages, II

• Prashant Pardeshi et al. NINJAL Parsed Corpus of Modern Japanese (NPCMJ) & Keyaki Treebank
  \( \approx 500K \) words

• Christina Tortora et al. The Audio-Aligned and Parsed Corpus of Appalachian English (AAPCApPE)
  \( \approx 1 \) million words

• Christina Tortora et al. A Corpus of New York City English (CUNY-CoNYCE).
  multi-million word corpus under construction
Coding queries
A canonical order sentence

( (IP-MAT (NP-SBJ (NPR John)))
  (VBP likes)
  (NP-OB1 (N pizza))
  (PUNC .)))
A topicalized sentence

( (IP-MAT (NP-OB1 (N Pizza)))
  (PUNC ,)
  (NP-SBJ (NPR John))
  (VBP likes)
  (PUNC .)))
A verb-second (V2) sentence

( (IP-MAT (NP-OB1 (N Pizza))
  (VBP likes)
  (NP-SBJ (NPR John))
  (PUNC .)))
A coding query example

node: IP-MAT*

ignore_nodes: PUNC | \\**

coding_query:

// grammatical status of first constituent coded in column 1

1: {

    subj: (IP-MAT* iDomsFirst NP-SBJ*)

    obj: (IP-MAT* iDomsFirst NP-OB1*)

    temp: (IP-MAT* iDomsFirst *-TMP)

    -: ELSE

}
Coding query, column 2

// position of finite verb

2: {
    \1: (IP-MAT* iDomsNum 1 finite_verb)
    \2: (IP-MAT* iDomsNum 2 finite_verb)
    \3: (IP-MAT* iDomsNum 3 finite_verb)
    -: ELSE
}

// subject-verb inversion?

3: {
    subj-fin: (IP-MAT* iDoms NP-SBJ*)
    AND (IP-MAT* iDoms finite_verb)
    AND (finite_verb precedes NP-SBJ*)

    fin-subj: (IP-MAT* iDoms NP-SBJ*)
    AND (IP-MAT* iDoms finite_verb)
    AND (NP-SBJ* precedes finite_verb)

    -: ELSE
}

Coding query, column 4

// status of subject

4: {

    pron: (IP-MAT* iDoms NP-SBJ*)
    AND (NP-SBJ* iDomsOnly PRO)

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

    -: ELSE

}
The canonical order sentence, coded

( (IP-MAT (CODING-IP-MAT subj : 2 : subj-fin : np)
  
  (NP-SBJ (NPR John))

  (VBP likes)

  (NP-OB1 (N pizza))

  (PUNC .)))
The topicalized sentence, coded

( (IP-MAT (CODING-IP-MAT obj : 3 : subj-fin : np)

  (NP-OB1 (N Pizza))

  (PUNC ,)

  (NP-SBJ (NPR John))

  (VBP likes)

  (PUNC .))))
The V2 sentence, coded

( (IP-MAT (CODING-IP-MAT obj : 2 : fin-subj : np)

  (NP-OB1 (N Pizza))

  (VBP likes)

  (NP-SBJ (NPR John)))

  (PUNC .)))
Extracting coding strings for quantitative analysis

Run a .q file with only the following single line:

```
  print_only: CODING*
```
Coding at more than one node

Sometimes it is useful to combine coding strings that CS generates at more node, for example at IP and at NP. It is possible to concatenate the strings into a single string.

This possibility requires the use of a function, called \texttt{concat} in the revision query module of CS, which we describe later on in this presentation.
The output for our toy example

subj : 2 : subj-fin : np

obj : 3 : subj-fin : np

obj : 2 : fin-subj : np
Some more realistic output

...
Importing coding strings into an R dataframe

<table>
<thead>
<tr>
<th>date</th>
<th>text</th>
<th>genre</th>
<th>clause</th>
<th>subj</th>
<th>fin</th>
<th>finord</th>
<th>nonfinord</th>
<th>fnonford</th>
<th>DO</th>
<th>IO</th>
<th>PPT</th>
<th>PPA</th>
<th>PPC</th>
<th>SOord</th>
<th>finDOord</th>
<th>nfDOord</th>
<th>DOquant</th>
<th>DOlen</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>0842</td>
<td>stras_p</td>
<td>p</td>
<td>rel</td>
<td>s_pro</td>
<td>main</td>
<td>s-fin</td>
<td>NONE</td>
<td>no-aux</td>
<td>o_tra</td>
<td>o_np</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>10</td>
<td>0842</td>
<td>stras_p</td>
<td>p</td>
<td>adv</td>
<td>s_Pro</td>
<td>mod-pouv</td>
<td>s-x-fin</td>
<td>s-non</td>
<td>non-fin</td>
<td>o-cli</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>s-o</td>
<td>o-fin</td>
<td>non-o</td>
<td>cli</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>0842</td>
<td>stras_p</td>
<td>p</td>
<td>rel</td>
<td>s_Dem</td>
<td>mod-pouv</td>
<td>s-x-fin</td>
<td>s-non</td>
<td>non-fin</td>
<td>o_tra</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>12</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>Mat</td>
<td>s_NP</td>
<td>main-etre</td>
<td>fin-s</td>
<td>NONE</td>
<td>no-aux</td>
<td>p_np</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>o-s</td>
<td>o-fin</td>
<td>NONE</td>
<td>ind</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>Mat</td>
<td>s_amb2</td>
<td>main-avoir</td>
<td>s-fin</td>
<td>NONE</td>
<td>no-aux</td>
<td>o_np</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>s-o</td>
<td>fin-o</td>
<td>NONE</td>
<td>conj</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>Mat</td>
<td>s_NP</td>
<td>mod-voul</td>
<td>fin-s</td>
<td>non-s</td>
<td>fin-non</td>
<td>o-cli</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>o-s</td>
<td>fin-o</td>
<td>o-non</td>
<td>cli</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>Mat</td>
<td>s_amb3</td>
<td>mod-voul</td>
<td>s-fin</td>
<td>s-non</td>
<td>fin-non</td>
<td>o_np</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>s-o</td>
<td>fin-o</td>
<td>non-o</td>
<td>def</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>Mat</td>
<td>s_Pro</td>
<td>main</td>
<td>s-fin</td>
<td>NONE</td>
<td>no-aux</td>
<td>o_np</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>s-o</td>
<td>fin-o</td>
<td>NONE</td>
<td>def</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>adv</td>
<td>s_Pro</td>
<td>main</td>
<td>s-x-fin</td>
<td>NONE</td>
<td>no-aux</td>
<td>o_np</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>s-o</td>
<td>o-fin</td>
<td>NONE</td>
<td>def</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>rel</td>
<td>s_tra</td>
<td>main</td>
<td>s-fin</td>
<td>NONE</td>
<td>no-aux</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>19</td>
<td>0900</td>
<td>eulal_p</td>
<td>p</td>
<td>Mat</td>
<td>s_NP</td>
<td>mod-pouv</td>
<td>s-fin</td>
<td>s-non</td>
<td>fin-non</td>
<td>o-cli</td>
<td>NONE</td>
<td>NONE</td>
<td>NONE</td>
<td>s-o</td>
<td>o-fin</td>
<td>o-non</td>
<td>cli</td>
<td>1</td>
</tr>
</tbody>
</table>
A case study: the rise of recipient passives in English (Bacovcin 2012)
Theme passives and recipient passives in Modern English

(1) John gave the books to Mary.
(2) The books were given to Mary (by John).
(3) John gave Mary the books.
(4) Mary was given the books (by John).
(5) *The books were given Mary (by John).
Theme passives and recipient passives in Modern German

(1) Hans gab der Maria den Artikel.
    DAT         ACC

(2) Der Artikel wurde der Mary (von Hans) gegeben.

(3) *Die Maria wurde den Artikel (von Hans) gegeben.

(4) Der Maria wurde der Artikel (von Hans) gegeben.
Ditransitive sentences in Early Middle English

(1) John gave Mary the book.
(2) John gave the book to Mary.
(3) John gave to Mary the book.
(4) John gave the book Mary.
Theme passives and recipient passives in Early Middle English

(1) The books were given to Mary (by John).
(2) The books were given Mary (by John).
(3) *Mary was given the book (by John).
German double accusatives

(1) Hans hat die Kinder Geschichte gelehrt.
   ACC ACC

(2) ?Hans hat den Kindern Geschichte gelehrt.
   DAT ACC

(3) *Geschichte wurde die Kinder gelehrt.

(4) Geschichte wurde den Kindern gelehrt.

(5) Die Kinder wurden Geschichte gelehrt.
Rise in the use of prepositional indirect objects in English

![Graph showing the rise in the use of prepositional indirect objects in English]
Rise in recipient passives in English
Markov Chain Monte Carlo simulations of the change

![Graph showing changes over time with various models and contexts]
Revision queries
Revision query 0.0: Concatenating coding strings

copy_corpus: t

query:  (NP* iDoms CODING-NP*)

AND (CODING-NP* iDoms [1]{2}.*)

AND (NP* iDoms CP-REL*)

AND (CP-REL* iDoms CODING-CP-REL*)

AND (CODING-CP-REL* iDoms [2]{1}.*)

concat{2, 1}:
English annotation for modals: Monoclausal structure

( (IP-MAT (NP-SBJ (PRO They)))

(MD will)

(VB come)

(ADVP-TMP (ADVR later)))))
Romance annotation for modals: Biclausal structure

( (IP-MAT (NP-SBJ (PRO They)))

(MD will)

(IP-INF (VB come))

(ADVP-TMP (ADVR later))))))
Revision query 1.0:
From monoclausal to biclausal structure

node: $ROOT

query: (IP-* iDoms MD)

AND (IP-* iDoms [1]{1}.*)

AND (MD iPrecedes [1].*)

AND (IP-* iDomsLast [2]{2}.*)

add_internal_node{1,2}: IP-INF
But what about punctuation?

( (IP-MAT (NP-SBJ (PRO They)))

  (MD will)

  (VB come)

  (ADVP-TMP (ADVR later)))

  (PUNC .)))
Revision query 1.1: Ignoring punctuation

node: $ROOT

ignore_nodes: PUNC

query: (IP-* iDoms MD)

   AND (IP-* iDoms [1]{1}.* )

   AND (MD iPrecedes [1].*)

   AND (IP-* iDomsLast [2]{2}.* )

add_internal_node{1,2}: IP-INF
Revision query 2:
From biclausal to monoclausal structure

node: $ROOT

query: (IP-* iDoms MD)

AND (IP-* iDoms {1}IP-INF)

AND (MD iPRecedes IP-INF)

delete_node{1}:
ECM annotation

( (IP-MAT (NP-SBJ (PRO They)))

(VBD saw)

(IP-INF (NP-SBJ (PRO him)))

(VB arrive))))
Accusativus cum infinitivo annotation

( (IP-MAT (NP-SBJ (PRO They)))
  (VBD saw)
  (NP-OBJ1 (PRO him))
  (IP-INF (VB arrive))))
Revision query 3.0:
From ECM to A.c.l.

node: $ROOT

query: (IP-* iDoms IP-INF)

   AND (IP-INF iDoms {1}NP-SBJ)

move_up_node{1}:

replace_label{1}: NP-OB1
Revision query 4.1:
From A.c.l. to ECM
	node: $ROOT

query: (IP-* iDoms {1}NP-OB1)

AND (IP-* iDoms {2}IP-INF)

AND (NP-OB1 iPrecedes IP-INF)

move_to{1,2}:

replace_label{1}: NP-SBJ
But we don’t want to revise cases of object control

( (IP-MAT (NP-SBJ (PRO They)))
  (VBD persuaded)
  (NP-OBJ1 (PRO him))
  (IP-INF (TO to)
    (VB come))))
Revision query 4.2: Restricting the revision to matrix “saw”

node: $ROOT

query: (IP-* iDoms {1}NP-OB1)

   AND (IP-* iDoms V*) AND (V* iDoms saw)

   AND (IP-* iDoms {2}IP-INF)

   AND (NP-OB1 iPrecedes IP-INF)

move_to{1,2}:

replace_label{1}: NP-SBJ
Revision query 4.3: Using iDomsMod

node: $ROOT

query:  
  (IP-* iDoms {1}NP-OB1)
  AND (IP-* iDomsMod V* saw)
  AND (IP-* iDoms {2}IP-INF)
  AND (NP-OB1 iPrecedes IP-INF)

move_to{1,2}:

replace_label{1}: NP-SBJ
A trivial definitions file

see: see* | saw
Revision query 4.4: Using the trivial definitions file

node: $ROOT

define: trivial.def

query: (IP-* iDoms {1}NP-OB1)

   AND (IP-* iDomsMod V* see)

   AND (IP-* iDoms {2}IP-INF)

   AND (NP-OB1 iPrecedes IP-INF)

move_to{1,2}:

replace_label{1}: NP-SBJ
A less trivial definitions file

feel: feel* | felt

hear: hear*

let: let*

see: see* | saw

ECM-verb: $feel | $hear | $let | $see
Revision query 4.5: Using the less trivial definitions file

node: $ROOT

define: less-trivial.def

query:  (IP-* iDoms {1}NP-OB1)
       AND (IP-* iDomsMod V* ECM-verb)
       AND (IP-* iDoms {2}IP-INF)
       AND (NP-OB1 iPrecedes IP-INF)

move_to{1,2}:

replace_label{1}: NP-SBJ
End