The Geneva Corpus of Middle English Poetry: its construction and possible applications

Richard Zimmermann

Université de Genève

November 16, 2013
Outline

Introduction
   What is the GeCMEP?
   Why is the GeCMEP useful?

Construction
   Step 1: Preprocessing
   Step 2: POS-tagging
   Step 3: Chunking

Database

Applications
   Example 1: Verb - Object order
   Example 2: Th and Wh elements
The Geneva Corpus of Middle English Poetry (GeCMEP) will be a fully annotated and syntactically parsed corpus. Its construction is a part of my PhD project.
The Geneva Corpus of Middle English Poetry (GeCMEP) will be a fully annotated and syntactically parsed corpus. Its construction is a part of my PhD project.

Time: 1150-1420 (Helsinki periods M1, M2, M3)
GeCMEP - overview

- The Geneva Corpus of Middle English Poetry (GeCMEP) will be a fully annotated and syntactically parsed corpus. Its construction is a part of my PhD project.
- Time: 1150-1420 (Helsinki periods M1, M2, M3)
- Size: goal is 100,000 words before end of PhD, but in principle open-ended
The Geneva Corpus of Middle English Poetry (GeCMEP) will be a fully annotated and syntactically parsed corpus. Its construction is a part of my PhD project.

- Time: 1150-1420 (Helsinki periods M1, M2, M3)
- Size: goal is 100,000 words before end of PhD, but in principle open-ended
- Parsed according to the rules of the PennParsed Corpus of Middle English (Kroch and Taylor 2000)
Example parse

(1) hwa swa ne for3efeô heore hating, ne god ne for3eueô him na þing
who so not forgives their hating, no God not forgives them no thing
'Whoever doesn't forgive their hate, God will not forgive them anything'
(PatNost,111.67.220)

(IP-MAT (NP-LFD (CP-FRL (WNP-1 (WPRO hwa) (ADV swa))
(C 0)
(IP-SUB (NP-SBJ *T*-1)
(NEG ne)
(VBP for+gefe+d)
(NP-OB1 (PRO$ heore) (N hating)))))

(, .)
(NP-SBJ (Q ne) (NPR god))
(NEG ne)
(VBP for+geue+d)
(NP-OB2-RSP (PRO him))
(NP-OB1 (Q na) (N +ting))
( . .))
(ID PatNost,111.67.220)
Introduction
Construction
Database
Applications

What is the GeCMEP?
Why is the GeCMEP useful?

Major Middle English Prose Texts 1100-1400

Substantial gap in prose record, 1250-1350!

1100 1150 1200 1250 1300 1350 1400

Time

1 In Festis Sancti Marie
2 Eluciderium
3 Anglo-Saxon Chronicle Continuation I
4 Anglo-Saxon Chronicle Continuation II
5 Lambeth Homilies
6 Trinity Homilies
7 Vespasian Homilies
8 Vices and Virtues
9 Ancrene Riwe
10 Katherine Group
11 Woolling Group
12 Kentish Sermons
13 Proclamation of Henry III
14 Aynhibe of Inwit
15 Richard Rolle
16 Earliest Prose Psalter
17 Mirror of Edmund
18 Travels of Sir Mandeville
19 Rievaulx' De Institutione
20 Wydiffe's works
21 Chaucer's Translation of Boethius
22 Trostic's Polychronicon
23 Julian of Norwich
24 Chaucer's Parson's Tale
25 Chaucer's Tale of Melibee
26 Texts from the Vernon Ms.
27 Chaucer's Treatise on the Astrolabe
28 The Cloud of Unknowing - author
29 Walter Hilton
30 The Chronicles of England
Major Middle English Prose Texts 1100-1400

→ ME verse can help to close the prose gap c. 1250-1350
Creation of a basic text file

- Find an electronic version of the text you want to parse

Some online resources...
Creation of a basic text file I

- Find an electronic version of the text you want to parse
- Example: Morris (1972) *An Old English Miscellany Containing a Bestiary*

Some online resources...

Go to Archive.org
Creation of a basic text file I

- Find an electronic version of the text you want to parse
- Example: Morris (1972) *An Old English Miscellany Containing a Bestiary*
  - Go to Archive.org
- Copy and paste into a .txt file with a formatting that can display special characters like thorn, yogh etc.

Some online resources...
Creation of a basic text file II

- Remove mark-ups, comments, footnotes, page & folio numbers, translations, critical apparatus etc. (UTF-8)

Replacement of ἐ in Notepad++
Creation of a basic text file II

- Remove mark-ups, comments, footnotes, page & folio numbers, translations, critical apparatus etc. (UTF-8)
- Replace special characters, e.g. $b \rightarrow +t$, $\omicron \rightarrow +g$ etc.

Replacement of $\omicron$ in Notepad++
Creation of a basic text file II

- Remove mark-ups, comments, footnotes, page & folio numbers, translations, critical apparatus etc. (UTF-8)
- Replace special characters, e.g. ḫ → þt, ζ → þg etc.
- Reformat the file such that there is only one word per line

Replacement of ô in Notepad++
Part-of-Speech Annotation

- POS-annotation with *Tree Tagger* (Schmid 1995)

**Example output of TreeTagger**

<table>
<thead>
<tr>
<th>word</th>
<th>pos</th>
<th>lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>DT</td>
<td>the</td>
</tr>
<tr>
<td>TreeTagger</td>
<td>NP</td>
<td>TreeTagger</td>
</tr>
<tr>
<td>is</td>
<td>VBZ</td>
<td>be</td>
</tr>
<tr>
<td>easy</td>
<td>JJ</td>
<td>easy</td>
</tr>
<tr>
<td>to</td>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>use</td>
<td>VB</td>
<td>use</td>
</tr>
<tr>
<td>.</td>
<td>SENT</td>
<td>.</td>
</tr>
</tbody>
</table>
Part-of-Speech Annotation

- POS-annotation with *Tree Tagger* (Schmid 1995)
- takes a word and assigns to it a POS-tag and a lemma

Example output of TreeTagger

<table>
<thead>
<tr>
<th>word</th>
<th>pos</th>
<th>lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>DT</td>
<td>the</td>
</tr>
<tr>
<td>TreeTagger</td>
<td>NP</td>
<td>TreeTagger</td>
</tr>
<tr>
<td>is</td>
<td>VBZ</td>
<td>be</td>
</tr>
<tr>
<td>easy</td>
<td>JJ</td>
<td>easy</td>
</tr>
<tr>
<td>to</td>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>use</td>
<td>VB</td>
<td>use</td>
</tr>
<tr>
<td>.</td>
<td>SENT</td>
<td>.</td>
</tr>
</tbody>
</table>
Part-of-Speech Annotation

- POS-annotation with *TreeTagger* (Schmid 1995)
- takes a word and assigns to it a POS-tag and a lemma
- GeCMEP does not include lemmas; lemmas not used

Example output of *TreeTagger*

<table>
<thead>
<tr>
<th>word</th>
<th>pos</th>
<th>lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>DT</td>
<td>the</td>
</tr>
<tr>
<td>TreeTagger</td>
<td>NP</td>
<td>TreeTagger</td>
</tr>
<tr>
<td>is</td>
<td>VBZ</td>
<td>be</td>
</tr>
<tr>
<td>easy</td>
<td>JJ</td>
<td>easy</td>
</tr>
<tr>
<td>to</td>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>use</td>
<td>VB</td>
<td>use</td>
</tr>
<tr>
<td>.</td>
<td>SENT</td>
<td>.</td>
</tr>
</tbody>
</table>
Part-of-Speech Annotation

- POS-annotation with *TreeTagger* (Schmid 1995)
- takes a word and assigns to it a POS-tag and a lemma
- GeCMEP does not include lemmas; lemmas not used
- supervised tagger; training lexicon and training input taken from PPCME2 prose texts

Example output of TreeTagger

<table>
<thead>
<tr>
<th>word</th>
<th>pos</th>
<th>lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>DT</td>
<td>the</td>
</tr>
<tr>
<td>TreeTagger</td>
<td>NP</td>
<td>TreeTagger</td>
</tr>
<tr>
<td>is</td>
<td>VBZ</td>
<td>be</td>
</tr>
<tr>
<td>easy</td>
<td>JJ</td>
<td>easy</td>
</tr>
<tr>
<td>to</td>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>use</td>
<td>VB</td>
<td>use</td>
</tr>
<tr>
<td>.</td>
<td>SENT</td>
<td>.</td>
</tr>
</tbody>
</table>
Part-of-Speech Annotation

- POS-annotation with *TreeTagger* (Schmid 1995)
- takes a word and assigns to it a POS-tag and a lemma
- GeCMEP does not include lemmas; lemmas not used
- supervised tagger; training lexicon and training input taken from PPCME2 prose texts
- accuracy: c. 85% of all tags are assigned correctly

Example output of TreeTagger

<table>
<thead>
<tr>
<th>word</th>
<th>pos</th>
<th>lemma</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>DT</td>
<td>the</td>
</tr>
<tr>
<td>TreeTagger is easy to use .</td>
<td>NP</td>
<td>TreeTagger</td>
</tr>
<tr>
<td>is</td>
<td>VBZ</td>
<td>be</td>
</tr>
<tr>
<td>easy</td>
<td>JJ</td>
<td>easy</td>
</tr>
<tr>
<td>to</td>
<td>TO</td>
<td>to</td>
</tr>
<tr>
<td>use</td>
<td>VB</td>
<td>use</td>
</tr>
<tr>
<td>.</td>
<td>SENT</td>
<td>.</td>
</tr>
</tbody>
</table>
Tokenization

- sentence boundaries and IDs inserted manually in a spreadsheet
Shallow parsing procedure

▶ shallow parsing builds simple syntactic structures with regular expressions (Abney 1991)
▶ e.g. prepositional phrases can be built with an instruction like "whenever there is a P immediately before an NP, then bracket them together into a PP"

```
  P  +  NP
      |        |
  for   D     N
      |        |
  our   God
```

```
  PP
  |
  P  +  NP
      |
  for  D     N
      |
  our   God
```
CorpusSearch revision queries

- tokens are chunked with revision queries of CorpusSearch 2 (Randall 2004)

**Example revision query**

```plaintext
define: cs.def
define: cs.def
node: IP*|CP*
copy_corpus: t
query: (IP*|CP* idoms {1}P)
AND (IP*|CP* idoms {2}NP)
AND (P iprecedes NP)
add_internal_node{1,2}: PP
```

- windows bat file runs very many of such revision queries; output of one query feeds into the next
- simple python script converts tokens into right input format
- then manual correction until all tokens correspond to PPCME2 guidelines
before chunking:

```
( IP-MAT (FW Natura)
  (FW leonis)
  (NUM ja.)
  (ID Bestiary, 1.1. FirstNature [Lion_Nature]))

( IP-MAT (D +De)
  (N leun)
  (VBP stant)
  (P on)
  (N hille)
  (ID Bestiary, 2.1.1. [Lion_Nature]))

( IP-MAT (P and)
  (PRO he)
  (N man)
  (VBP hurten)
  (VBP here)
  (CONJ or-der)
  (VP +durg)
  (PRO pi)
  (N nese)
  (N smel)
  (VBP Smake)
  (C +dat)
  (PRO he)
  (VBP negge)
  (P B1)
  (VBP mit)
  (N wele)
  (ADV so)
  (PRO he)
  (MD while)
  (P TO)
  (N dele)
  (ADV mt-der)
  (VBP wenden)
  (ID Bestiary, 3.1.6. [Lion_Nature]))
```

after chunking:

```
( (IP-MAT (FW Natura)
  (FW leonis)
  (NUM ja.)
  (ID Bestiary, 1.1. FirstNature [Lion_Nature]))

( (IP-MAT (D +De)
  (N leun)
  (VBP stant)
  (PP (P on)
    (NP (N hille))
  )
  (ID Bestiary, 2.1.1. [Lion_Nature]))

( (IP-MAT (P and)
  (CP-ADV (C 0)
    (IP-SUB (NP-SBJ (PRO he))
      (NP-OB1 (N man))
      (VP hurten)
      (VBP hure)
      (PP (P +durg)
        (NP (PRO pi) (N nese))
        (NP-SBJ (N smel))
      )
      (VBP Smake)
      (CP (C +dat)
        (IP-SUB (NP-SBJ (PRO he))
          (VBP negge)
          (PP (P B1)
            (CP-ADV (C 0)
              (IP-SUB (MD wilc)
                (NP-OB1 (N wele))
                (ADVP (ADV so))
                (NP-SBJ (PRO he))
                (MD wilc)
                (PP (P TO)
                  (NP (N dele))
                  (ADVP (ADV mt-der))
                  (VBP wenden)))))
```
A more efficient workflow?

- The corpus construction process is slow and tedious.
A more efficient workflow?

- The corpus construction process is slow and tedious.
- I will try to learn how to optimize the construction process from experts at the University of Pennsylvania during a stay 2014-2015.
Currently the database includes information on 15 parsed text files and a general bibliography.
Currently the database includes information on 15 parsed text files and a general bibliography.

Text information specifies factors that might determine syntactic variation: date of composition, date of manuscript, dialect, versification, literary subjects.
Currently the database includes information on 15 parsed text files and a general bibliography.

Text information specifies factors that might determine syntactic variation: date of composition, date of manuscript, dialect, versification, literary subjects.

In addition, cross-references to the three standard catalogues of ME (verse) texts: IMEV, Wells and MEC.
Currently the database includes information on 15 parsed text files and a general bibliography.

Text information specifies factors that might determine syntactic variation: date of composition, date of manuscript, dialect, versification, literary subjects.

In addition, cross-references to the three standard catalogues of ME (verse) texts: IMEV, Wells and MEC.
Searching the corpus

- The corpus can be searched with query files of CorpusSearch 2 (Randall 2004)
Searching the corpus

- the corpus can be searched with query files of CorpusSearch 2 (Randall 2004)
- all functional labels, POS-tags and specific spellings can be searched for
Searching the corpus

- the corpus can be searched with query files of CorpusSearch 2 (Randall 2004)
- all functional labels, POS-tags and specific spellings can be searched for
- complex relations between elements can be specified (relative order of elements, number of words, identical indices, ...)

Realization of *verb-pronoun* structures

(2)  

a. þe þurst **him** dede more wo  
the thirst him did more woe  
þen hevede ræper his hounger do.  
than had earlier his hunger done  
'
'The thirst caused him more misery than his hounger had done before' (FoxWolf,56.273.68)

b. þe wox **bicharde** **him**, mid iwisse,  
the fox deceived him, with certainty,  
For he ne fond nones kunnes blisse  
for he not found none kind’s bliss  
'
'The fox had deceived him indeed, because he didn’t find any kind of bliss’ (FoxWolf,224.278.295)
Search queries for *verb-pronoun* structures

- simple CorpusSearch query files to find such constructions:

**opro-V.q**

define: cs.def
node: IP*
query: (IP* idoms NP-OB*)
AND (NP-OB* idomsonly PRO)
AND (IP* idoms VBP|VBD|DOP|DOD|HVP|HVD)
AND (NP-OB* precedes VBP|VBD|DOP|DOD|HVP|HVD)

**V-opro.q**

define: cs.def
node: IP*
query: (IP* idoms NP-OB*)
AND (NP-OB* idomsonly PRO)
AND (IP* idoms VBP|VBD|DOP|DOD|HVP|HVD)
AND (VBP|VBD|DOP|DOD|HVP|HVD precedes NP-OB*)
Development of verb-pronoun structures

→ Significant decline of object pronoun - verb orders (c. 80% to c. 40%) measurable in Middle English verse 1150-1400
Benefits

- Collecting data from parsed corpora with automated search queries ...
  - ... saves a lot of time. Going through tens of thousands of words manually takes weeks or even months - with parsed corpora it’s a matter of seconds.
Benefits

- Collecting data from parsed corpora with automated search queries ...
  - ... saves a lot of time. Going through tens of thousands of words manually takes weeks or even months - with parsed corpora it’s a matter of seconds.
  - ... causes fewer mistakes. Humans can easily overlook examples or tally them up wrong - computers count correctly.
Benefits

- Collecting data from parsed corpora with automated search queries ...
  - ... saves a lot of time. Going through tens of thousands of words manually takes weeks or even months - with parsed corpora it’s a matter of seconds.
  - ... causes fewer mistakes. Humans can easily overlook examples or tally them up wrong - computers count correctly.
  - ... assures replicability. Researchers can quickly double-check quantitative claims in the literature.
Collecting data from parsed corpora with automated search queries ...

- ... saves a lot of time. Going through tens of thousands of words manually takes weeks or even months - with parsed corpora it’s a matter of seconds.
- ... causes fewer mistakes. Humans can easily overlook examples or tally them up wrong - computers count correctly.
- ... assures replicability. Researchers can quickly double-check quantitative claims in the literature.
- ... increases objectivity. Search criteria must be made explicit in the query files hence the ex-/inclusion of a particular sentence is independent of the individual researcher.
Benefits

- Collecting data from parsed corpora with automated search queries ...
  - ... saves a lot of time. Going through tens of thousands of words manually takes weeks or even months - with parsed corpora it’s a matter of seconds.
  - ... causes fewer mistakes. Humans can easily overlook examples or tally them up wrong - computers count correctly.
  - ... assures replicability. Researchers can quickly double-check quantitative claims in the literature.
  - ... increases objectivity. Search criteria must be made explicit in the query files hence the ex-/inclusion of a particular sentence is independent of the individual researcher.

→ increased scientificity
Locative relatives: there → where ...

(3) 

a. his halie nome we nomen and beren. 
his holy name we took and bore 
In þe font þer we iclensed weren. 
in the font where we cleansed were 
'Ve took and bore his name in the font where we were cleansed.' (PatNost, 36.59.70)

b. To Oxenford is messageris he sende, that hi soghte 
to Oxford his messengers he sent, that they sought 
This maide ware heo were ifounde and to him broghte. 
this maid where she were found and to him brought 
'He sent his messengers to Oxford so that they might seek and find this maiden, where she was, and bring her to him.' (Fridesw, 47.64)
and temporal subordinators: *then* $\rightarrow$ *when*

(4)  a. Ac ure drihten eft of deaþe hem arærep, but our Lord again of death them arises
    So he alle men deþ, *bonne* domes dai cumeþ.
    as he all men does when Doomsday comes
    ’But our Lord will raise them up from death, as he will all men, when Doomsday comes.’
    (BodySoul,185.7.13.FragE)

b.  and al bi-fuliþ he his frend
    and all befouls he his friend
    *hwæn* he him vnfoldiþ.
    when he him embraces
    ’He wholly befouls his friend when he embraces him.’
    (ProvAlf,224.50.659.B32)
→ Significant increase of *Wh*- relative and adverbial clauses (c. 10% to c. 70%) in Middle English verse 1150-1400
Comparison verse - prose ...
... shows that verse can close the gap in prose texts.
Special thanks to Beatrice Santorini for running her queries on the GeCMEP files to find annotation mistakes. Thanks are also due to Benjamin Börschinger and Paola Merlo for useful advice on chunking.


Randall, B. (2004), *CorpusSearch 2*,