Syllabification of Middle Dutch

Gosse Bouma ¹  Ben Hermans ²

¹Centre for Language and Cognition
University of Groningen

²Meertens Institute
Amsterdam

ACRH-2, Nov 29, 2012
Syllabification of Historical Texts

Motivation

- Understand phonological processes and differences in the dialects of Middle Dutch
- Spelling variation may correlate with phonological variation
  - priest: *priester, preyster, prester*
- Word tokens only provide information about word initial/internal/final phenomena
- Syllabified tokens can also be studied on the level of onset/nucleus/coda
  - i.e. distribution of *gh* in onset vs. coda position

Challenges

- Spelling variation: some letters or graphemes can stand for vowels as well as consonants
- Lack of gold standard data for development and evaluation
Syllabification of Historical Texts

Motivation

- Understand phonological processes and differences in the dialects of Middle Dutch
- Spelling variation may correlate with phonological variation
  - priest: *priester, preyster, prester*
- Word tokens only provide information about word initial/internal/final phenomena
- Syllabified tokens can also be studied on the level of onset/nucleus/coda
  - i.e. distribution of *gh* in onset vs. coda position

Challenges

- **Spelling variation**: some letters or graphemes can stand for vowels as well as consonants
- **Lack of gold standard data** for development and evaluation
Finite-State Hyphenation

If an NLP problem can be conveniently solved with a finite-state automaton, then it is probably a good idea to solve it that way

Gazdar and Mellish, 1989

Natural Language Processing in Prolog/Pop11/Lisp

Approach

- Manual definition of a FST to do hyphenation with reasonable accuracy
- Annotate sample of word types from the corpus using the baseline FST
- Learn error-correcting rules using Transformation-based Learning
- Compose result into a single FST with high accuracy
Finite-State Hyphenation

If an NLP problem can be conveniently solved with a finite-state automaton, then it is probably a good idea to solve it that way

Gazdar and Mellish, 1989
Natural Language Processing in Prolog/Pop11/Lisp

Approach

- Manual definition of a FST to do hyphenation with reasonable accuracy
- Annotate sample of word types from the corpus using the baseline FST
- Learn error-correcting rules using Transformation-based Learning
- Compose result into a single FST with high accuracy
The Corpus van Reenen-Mulder
The Corpus van Reenen-Mulder

Wy borgchemestere ende raet van Groningen doet cundich al denghenen, de dessen openen bref sien of horen lesen, dat Sybrant Meynckens quam vor uns ende bikande des, dat he Johanne van Buren ende Byen zinen wive hadde vercoft een hues, dat staet in der Cremerijp, umme ene summe van ghelde, de em vol ende al bitalet.

We mayer and counsil of Groningen announce to everyone who see or hear read this letter, that Sybrant Meynkens came to us and stated that he had sold to Johan van Buren and his wife a house, located in Kropswolde for an amount of money that has been paid fully.

Composition

- 2.700 legal texts, 14th century, from the Netherlands (including Flanders)
- Metadata: date, location
- Transcripts available in electronic form
Hyphenating Modern Dutch

Selecting Correct Syllabification

1. Syllable boundaries cannot cross morpheme boundaries.
2. A syllable consists of a sequence of well-formed (onset) nucleus (coda)
3. **Maximum onset principle**: between syllables, consonants are added to the onset if possible

<table>
<thead>
<tr>
<th>drugspanden</th>
<th>Morpheme</th>
<th>Syllable</th>
<th>MaxOnset</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>√</td>
<td>√</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

Gosse Bouma, Ben Hermans
Hyphenating Modern Dutch

Selecting Correct Syllabification

1. Syllable boundaries cannot cross morpheme boundaries.
2. A syllable consists of a sequence of well-formed (onset) nucleus (coda)
3. Maximum onset principle: between syllables, consonants are added to the onset if possible

<table>
<thead>
<tr>
<th>drugspanden (drugs-houses)</th>
<th>Morpheme</th>
<th>Syllable</th>
<th>MaxOnset</th>
</tr>
</thead>
<tbody>
<tr>
<td>drug-span-den</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>drugs-pa-nden</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>drugs-pand-en</td>
<td>√</td>
<td>√</td>
<td>**</td>
</tr>
<tr>
<td>drugs-pan-den</td>
<td>√</td>
<td>√</td>
<td>*</td>
</tr>
</tbody>
</table>
Finite-state Hyphenation

Karttunen 1998: Optimality Theory as FST
- Use lenient composition to model OT constraints:
  - GEN .O. HaveOns .O. NoCoda .O. FillNuc ...

Karttunen 1995: Replace-operator
- replace (In x Out, LC, RC)
- replace the longest string matching In by Out in context LC _ RC
FST Hyphenation (Bouma 2003)

**Approach**

- `aardappel`
- `+aardappel`
- `+@aa@rd@a@pp@e@l`
- `+@aa@r@-d@a@p@-p@e@l`
- `aar-dap-pel`

Mark the beginning of a word
Mark the beginning and end of each nucleus
Insert hyphen before each maximal onset
Remove markers

**Limitations**

- Requires manual, language specific, definition of onset and nucleus
- Ignores morpheme boundaries
FST Hyphenation (Bouma 2003)

Approach

- aardappel
- \( +\text{aardappel} \)
- \( +\text{aa}@\text{r-d}@\text{a}@\text{p}@\text{p}@\text{e}@\text{l} \)
- \( +\text{aa}@\text{r-d}@\text{a}@\text{p-p}@\text{e}@\text{l} \)
- aar-dap-pel

- Mark the beginning of a word
- Mark the beginning and end of each nucleus
- Insert hyphen before each maximal onset
- Remove markers

Limitations

- Requires manual, language specific, definition of onset and nucleus
- Ignores morpheme boundaries
Implementation

FSA Utilities (Gerdemann and van Noord, 1999)

markNucleus
replace( nucl × id(nucl) @ , ε, ε ) @aa@r@d@e@

insertHyphens
replace( ε × - , [@, cons*], [onset?, @] ) @aa@r-d@e@

Nucleus: a, aai, au, e, ee, ei, ...
Onset: b, bl, br, c, ch, chr, cl, cr, d, dr, dw, ...

Evaluation

- word list from CELEX (avg. 2.5 hyphens per word)
- Major source of errors: compounds
- Accuracy can be improved substantially with Transformation-based learning

<table>
<thead>
<tr>
<th></th>
<th>FST</th>
<th>TBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>word accuracy</td>
<td>86.0%</td>
<td>98.5%</td>
</tr>
<tr>
<td>hyphenation accuracy</td>
<td>94.0%</td>
<td>99.1%</td>
</tr>
</tbody>
</table>
Implementation

FSA Utilities (Gerdemann and van Noord, 1999)

markNucleus

\[
\text{replace}( \text{nucl} \times @ \text{id(nucl)} @ , \epsilon, \epsilon ) \quad \text{aa}r\text{d}e
\]

insertHyphens

\[
\text{replace}( \epsilon \times -, [@, \text{cons*}], [\text{onset？}, @] ) \quad \text{aa}r-d\text{e}
\]

Nucleus: a, aai, au, e, ee, ei, ...
Onset: b, bl, br, c, ch, chr, cl, cr, d, dr, dw, ...

Evaluation

- word list from CELEX (avg. 2.5 hyphens per word)
- Major source of errors: compounds
- Accuracy can be improved substantially with Transformation-based learning

<table>
<thead>
<tr>
<th></th>
<th>FST</th>
<th>TBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>word accuracy</td>
<td>86.0%</td>
<td>98.5%</td>
</tr>
<tr>
<td>hyphenation accuracy</td>
<td>94.0%</td>
<td>99.1%</td>
</tr>
</tbody>
</table>
Challenges for Middle Dutch

(chaplain) capelaen, capellaen, kapelaen, capelane, capellanen
(church yard) kerchoef, kerchoe, kerkhoue, kirchoef, kirchove
(mayor) borgchermestere, borghermeester, borghermeistere, borghermeyster, burchmeester
& 19 more
## Challenges

### Ambiguous graphemes

<table>
<thead>
<tr>
<th>G1</th>
<th>G2</th>
<th>example</th>
<th>example</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>j</td>
<td>iaer</td>
<td>jaer</td>
<td>year</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ighelic</td>
<td>jghelic</td>
<td>in fact</td>
</tr>
<tr>
<td>u</td>
<td>v</td>
<td>uerclaringhen</td>
<td>verclaringhen</td>
<td>statements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uerstaen</td>
<td>verstaen</td>
<td>understand</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zeuentien</td>
<td>zeventien</td>
<td>seventeen</td>
</tr>
<tr>
<td>uu</td>
<td>w</td>
<td>uuutghesproken</td>
<td>wtghesproken</td>
<td>stated</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zuutzide</td>
<td>zwtzide</td>
<td>southside</td>
</tr>
<tr>
<td>uu</td>
<td>vv</td>
<td>huus</td>
<td>hvvs</td>
<td>house</td>
</tr>
</tbody>
</table>

Gosse Bouma, Ben Hermans

11/19
Dealing with ambiguous graphemes

New Symbols

\[
\begin{align*}
U &= u \text{ functioning as consonant} \\
V &= v \text{ functioning as vowel} \\
W &= w \text{ functioning as vowel} \\
J &= j \text{ functioning as vowel}
\end{align*}
\]

Preprocessing Rules

\[
\begin{align*}
\text{aue, eue, oui} & \rightarrow aUe, eUe, oUi \\
C_1 vC_2 + vC_2 & \rightarrow C_1 VC_2 + VC_2 \quad (C_n = \text{a consonant}) \\
C_1 wC_2 + wC_2 & \rightarrow C_1 WC_2 + WC_2 \quad (C_n = \text{a consonant}) \\
+jn, +jm & \rightarrow +Jn, +Jm
\end{align*}
\]

Definition of nucleus and onset extended with new symbols
Creating a gold standard

Data
Rule-based finite-state hyphenator was used to automatically hyphenate all word types in the corpus
50% manually corrected (20K word types)

Performance of rule-based system

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>word accuracy</td>
<td>90.1%</td>
</tr>
<tr>
<td>hyphenation accuracy</td>
<td>94.0%</td>
</tr>
</tbody>
</table>
Creating a gold standard

Data

Rule-based finite-state hyphenator was used to automatically hyphenate all word types in the corpus. 50% manually corrected (20K word types)

Performance of rule-based system

- word accuracy: 90.1%
- hyphenation accuracy: 94.0%
**Transformation-based Learning**

Brill 1995, Ngai & Florian 2001

Given aligned gold standard data and system output, learn optimal rules to correct frequent errors

Alignment

<table>
<thead>
<tr>
<th>word</th>
<th>(answered)</th>
<th>and-wer-de</th>
<th>0 0 1 0 0 0 1 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>FST</td>
<td>and-wer-de</td>
<td>0 0 2 0 0 0 1 0</td>
<td></td>
</tr>
<tr>
<td>correct</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>word</td>
<td>(pilgrimage)</td>
<td>be-deu-aerd</td>
<td>0 0 1 0 0 1 0 0 0</td>
</tr>
<tr>
<td>FST</td>
<td>be-de-uaerd</td>
<td>0 0 1 0 0 9 0 0 0</td>
<td></td>
</tr>
<tr>
<td>correct</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Transformation-based Learning

Brill 1995, Ngai & Florian 2001

Given aligned gold standard data and system output, learn optimal rules to correct frequent errors

Alignment

0 = no hyph, 1 = preceded by hyph, 2 = followed by hyph, 9 = prev char preceded by hyph

<table>
<thead>
<tr>
<th>word</th>
<th>(answered)</th>
<th>a n d w e r d e</th>
</tr>
</thead>
<tbody>
<tr>
<td>FST</td>
<td>an-dwer-de</td>
<td>0 0 1 0 0 0 1 0</td>
</tr>
<tr>
<td>correct</td>
<td>and-wer-de</td>
<td>0 0 2 0 0 0 1 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>word</th>
<th>(pilgrimage)</th>
<th>b e d e v a e r d</th>
</tr>
</thead>
<tbody>
<tr>
<td>FST</td>
<td>be-deu-aerd</td>
<td>0 0 1 0 0 1 0 0 0</td>
</tr>
<tr>
<td>correct</td>
<td>be-de-uaerd</td>
<td>0 0 1 0 9 0 0 0 0</td>
</tr>
</tbody>
</table>
## Transformation-based Learning

### Rules

Over 100 error correcting rules learned, most important:

- `-ster` → `s-ter`
- `-ru` → `r-u`
- `-fl` → `f-l`
- `-y` → `y`
- `eu-a` → `e-ua`
- `+io-` → `+io`
- `c-hei` → `ch-ei`
- `l-ue` → `lu-e`

### Accuracy (10-fold cross validated)

<table>
<thead>
<tr>
<th></th>
<th>FST</th>
<th>TBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>word accuracy</td>
<td>90.1</td>
<td>96.5% (s.d. = 0.317)</td>
</tr>
<tr>
<td>hyphenation accuracy</td>
<td>94.0</td>
<td>97.9% (s.d. = 0.173)</td>
</tr>
</tbody>
</table>

### Compilation

Rule-based and TBL rules can be combined into efficient and small FST (131 states, 442 kB)
Transformation-based Learning

Rules

Over 100 error correcting rules learned, most important:
- ster → s-ter
- ru  → r-u
- fl  → f-l
- y   → y

eu-a → e-ua
+io- → +io
c-hei → ch-ei
l-ue → lu-e

Accuracy (10-fold cross validated)

<table>
<thead>
<tr>
<th></th>
<th>FST</th>
<th>TBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>word accuracy</td>
<td>90.1</td>
<td>96.5%</td>
</tr>
<tr>
<td>hyphenation accuracy</td>
<td>94.0</td>
<td>97.9%</td>
</tr>
</tbody>
</table>

(s.d. = 0.317)

(s.d. = 0.173)

Compilation

Rule-based and TBL rules can be combined into efficient and small FST (131 states, 442 kB)
Transformation-based Learning

Rules

Over 100 error correcting rules learned, most important:
- ster → s-ter
eu-a → e-ua
- ru → r-u
+io- → +io
- fl → f-l
c-hei → ch-ei
- y → y
l-ue → lu-e

Accuracy (10-fold cross validated)

<table>
<thead>
<tr>
<th></th>
<th>FST</th>
<th>TBL</th>
</tr>
</thead>
<tbody>
<tr>
<td>word accuracy</td>
<td>90.1</td>
<td>96.5%</td>
</tr>
<tr>
<td>hyphenation accuracy</td>
<td>94.0</td>
<td>97.9%</td>
</tr>
</tbody>
</table>

(s.d. = 0.317) (s.d. = 0.173)

Compilation

Rule-based and TBL rules can be combined into efficient and small FST (131 states, 442 kB)
ei vs. ey

Distribution (number of occurrences per 1000 syllables) of the nuclei ei and ey over time
gh vs. g onsets

Distribution of the onsets gh and g over time

Gosse Bouma, Ben Hermans
Geographical spread

Percentage $\text{ei}$ of total $\text{ei} + \text{ey}$ occurrences

Dark = more $\text{ei}$

Map created using www.gabmap.nl
Conclusions

Results

- Adapted finite-state hyphenation method for modern Dutch to Middle Dutch
- Improved using Transformation-based learning
- 97.9% hyphenation accuracy
- Fast implementation
- Hyphenated complete corpus, preliminary exploration of the data

Future Work

- More principled phonological research
- Applying method to other data (e.g. Corpus Gysseling, 13th century Dutch)
Conclusions

Results

- Adapted finite-state hyphenation method for modern Dutch to Middle Dutch
- Improved using Transformation-based learning
- 97.9% hyphenation accuracy
- Fast implementation
- Hyphenated complete corpus, preliminary exploration of the data

Future Work

- More principled phonological research
- Applying method to other data (e.g. Corpus Gysseling, 13th century Dutch)