UPPSALA UNIVERSITET

NLP for Historical (or Very Modern) Text

Eva Pettersson

eva.pettersson@lingfil.uu.se

2017-08-30

Aims and Motivation

- Historical text constitutes a rich source of information
- Not easily accessed
- Many texts are not digitized
- Lack of language technology tools to handle even digitized historical text
- Leads to time-consuming manual work for historians, philologists and other researchers in humanities

UPPSALA UNIVERSITET

Example: Gender and Work

- Historians are interested in what man and women did for a living in the Early Modern Swedish Society (appr. 1550—1800)
- Information stored in database
- Often expressed as verb phrases

hugga ved	'chop wood'
sälja fisk	'sell fish'
tjäna som piga	'serve as a maid'

LT Solution for the GaW Project

1. Automatic extraction of verb phrases from historical text, based on tagging and parsing

2. Statistical methods for automatic ranking of the extracted phrases to display phrases describing work at the top of the results list

(Some) Challenges with Historical Text

- Different and inconsistent spelling
- Different vocabulary (often with Latin influences)
- Different (and inconsistent) morphology
- Longer sentences
- Inconsistent use of punctuation
- Different syntax and inconsistent word order
- Code-switching
- Substantial differences between texts from different time periods, genres, and authors

Spelling

- Both diachronic and synchronic spelling variance
- Lack of spelling conventions
- Spell the way words sound – different dialects
- Spellings of pronoun mig (‘me/myself’) in the Swedish book of prayers *Svenska tideboken* (1525):

mig
migh
mik
mic
mich
mech

Spelling Variation Extreme

- The word tiuvel (Teufel) ‘devil’ occurs 733 times in *Reference Corpus of Middle High German* with 90 different spellings:

  dievel diuel diufal diuual diu=uil diuvil divel divuel divuil divvel dufel duoifel duovel duuel duuil duvel duvil dvofel dvuil dwowel lieuel loufel teufel tevfel thufel thuuil tiefal tiefel tiefil tieuel tie=uel tieuil tieuuel tieuuil tievel ti=evel tie=vel tievil tifel tiofel tiuel tiufal tiufel tiufil tiufle tiuil tiuofel tiuuel tiuuil tiuval tiuvel tiuvil tivel tivfel tivil tivuel tivuil tivvel tivvil tivwel tiwel tubel tubil tueuel tufel tufil tuifel tuofel tuouil tuovel tuovil tuuel tuuil tuujl tuvel tuvil tvfel tvivel tvivil tvouel tvouil tvovel tvuel tvuil tvvel tvvil tyefel tyeuel tyevel tyfel

Vocabulary

- New words enter the language (e.g., technological development)
- Old words become less frequent or eventually non-existing
- Early New High German Words (1350–1650) not in use today*:

liberei/librari	Bibliothek	'library'
triangel	Dreieck	'triangle'
akkord	Vertrag	'treaty'

* Salmons (2012): *A History of German – What the past reveals about today's language*

Morphology

- Analogical levelling
- Shift in inflection from strong to weak paradigm

Historical English	Modern English*
old – elder – eldest	old – older – oldest

Martin Luther (1483–1546)	Modern German*
er bleyb/sie blieben er fand/sie funden	er blieb/sie blieben er fand/sie fanden

* Campbell (2013): *Historical linguistics*

Syntax

- Word order differences
- English transforming from synthetic language to (mostly) analytic language
- Synthetic languages
  - Highly inflected
  - Word endings mark grammatical functions
  - Less strict word order
- Analytic languages
  - Fewer word endings
  - Word order important clue for interpreting the grammatical functions of the words in a sentence

Sentence Boundaries and Sentence Length

- Not trivial to determine where one sentence ends and another sentence begins:
  - full stop succeeded by uppercase letter
  - full stop not succeeded by uppercase letter
  - slash, comma, semi-colon or other sign to mark sentence boundaries (with or without succeeding uppercase letter)
  - uppercase letter without preceding punctuation mark
  - no sentence boundary marker at all…
- Sentence boundary strategy may vary throughout the same document

How to Tag and Parse Historical Text?

Two main approaches:

1. Train a tagger/parser on historical data
   - Data sparseness issues

2. Spelling Normalisation
   - Automatically translate the original spelling to a more modern spelling, before performing tagging and parsing
   - Enables the use of NLP tools available for the modern language
   - Does not take into account syntactic differences, and changes in vocabulary

Spelling Normalisation

- Rule-based Normalisation
- Levenshtein-based Normalisation*
  - Edit distance comparisons between the historical word form and a modern dictionary or corpus
- Memory-based Normalisation*
  - Parallel corpus of token pairs with historical spelling mapped to modern spelling
- SMT-based Normalisation*

* Evaluated and compared in Pettersson et al. (2014): *A Multilingual Evaluation of Three Spelling Normalisation Methods for Historical Text*

Rule-based Normalisation

- Hand-written normalisation rules based on known language changes and/or empirical findings
- Swedish examples:
  - drop of the letters *-h* and *-f* for the *v* sound

    hvar → var 'was'

    skrifva → skriva 'write'
  - deletion of repeated vowels

    saak → sak 'thing'
  - substitution of phonologically similar letters

    qvarn → kvarn 'mill'

    slogz → slogs 'were fighting'

Levenshtein-based Normalisation

- Edit distance comparisons between the historical word form and word forms present in a modern dictionary or corpus
- The word form in the dictionary that is most similar to the historical word form is chosen, if the similarity is large enough
- Weighted edit distance, taking into account known spelling changes, could boost the performance

Levenshtein-based Normalisation

Edit distance comparisons between the historical word form and tokens present in a modern dictionary/corpus

ryghtful

rightful

Levenshtein-based Normalisation

Edit distance comparisons between the historical word form and tokens present in a modern dictionary/corpus

ryghtful

rightful

1 substitution

Levenshtein-based Normalisation

Edit distance comparisons between the historical word form and tokens present in a modern dictionary/corpus

ryghtful

rightful

1 substitution =
edit distance 1

Memory-based Normalisation

- Parallel training corpus of word form pairs with historical spelling mapped to modern spelling
- Most frequent equivalent is chosen ≈ dictionary lookup

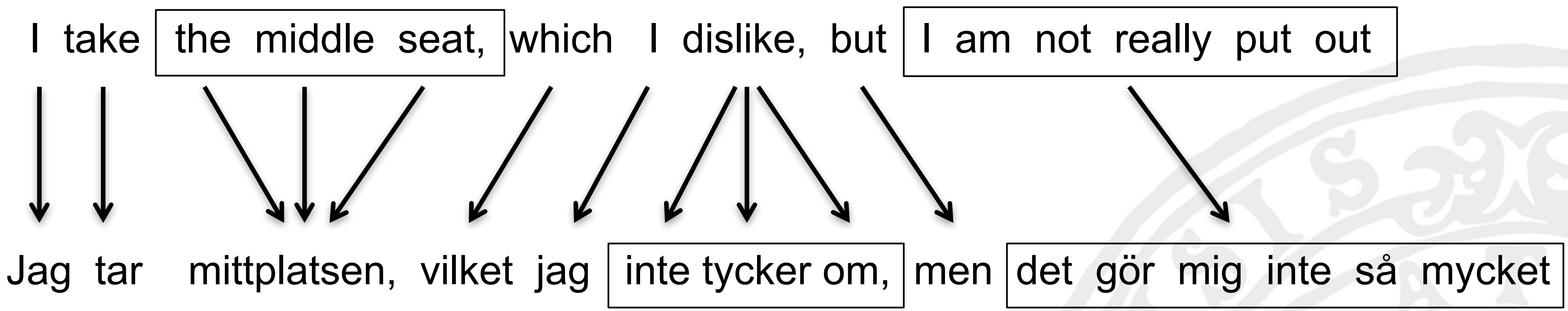
moost	most
noble	noble
&	and
worthiest	worthiest
lordes	lords
moost	most
ryghtful	rightful
conseille	council

SMT-based Normalisation

- Spelling normalisation treated as a translation task
- Standard Moses settings using GIZA++
- Translation based on character sequences rather than words and phrases*
- Previously performed for translation between closely related languages
- Only small parallel corpus needed for training due to fewer possible combinations of characters than of words

*Further described in Pettersson et al. (2013): *An SMT Approach to Automatic Annotation of Historical Data*

SMT Word Alignment



Normalisation Character Alignment

m $\boxed{\text{o o}}$ s t

m o s t

Very Modern Data

- The same methods that are used for NLP for historical text have also been used for very modern text, such as Twitter data
- Spelling normalisation useful before tagging/parsing

seein that ad makes me wanna listen to dat song rite now

Example from Clark & Araki (2011)

Suggestions for Projects

1. Spelling Normalisation

- Aim:
  - developing your own system for spelling normalisation of historical text, or modern data such as Twitter data

- Possible methods:
  - manually or automatically defined re-write rules
  - (Levenshtein) edit distance comparisons
  - phonetic similarity
  - statistical machine translation techniques
  - neural network techniques
  - …or any method you can come up with!
    (including combinations of different approaches)

Suggestions for Projects

2. Tagging and Parsing

- Aim:
  - developing methods for tagging and/or parsing of historical text, or modern data such as Twitter data

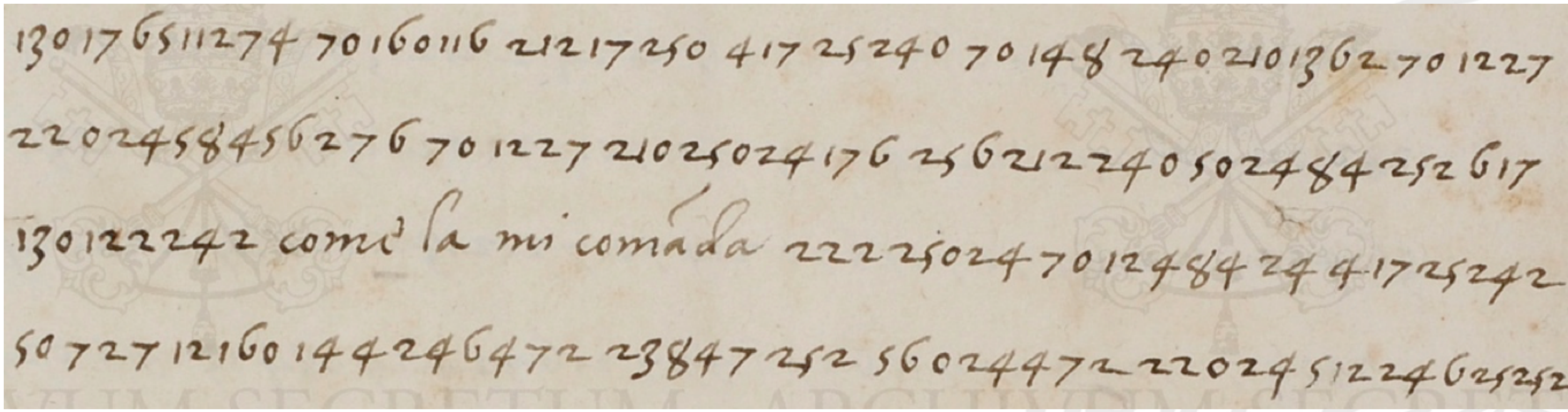
- Challenge:
  - take into account the special characteristics of historical/Twitter text, such as orthographic and syntactic variance

Suggestions for Projects

3. Detecting Cleartext in a Cipher

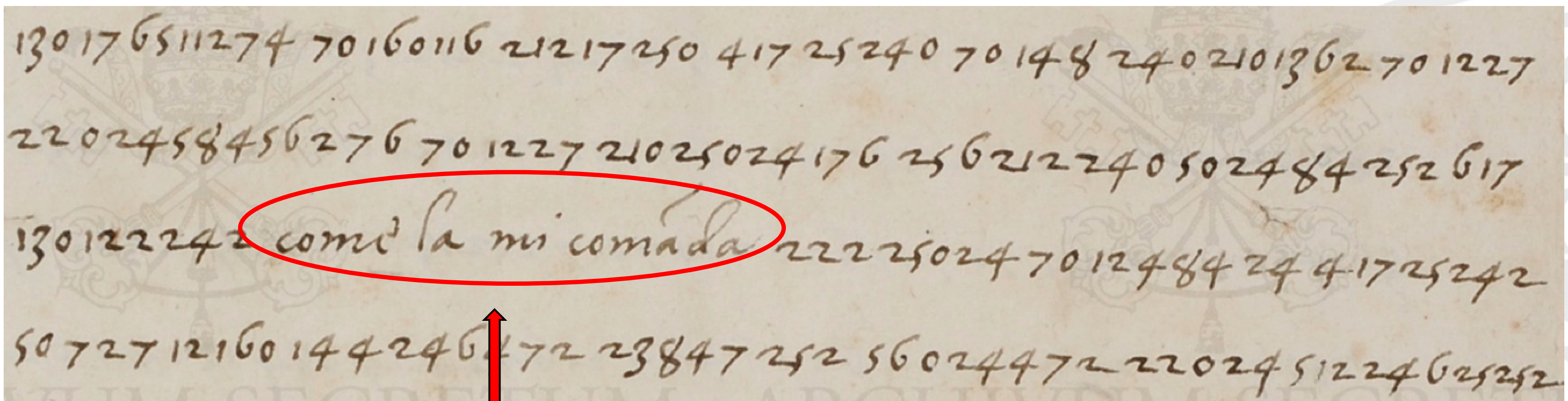
- Historical ciphers are encoded, hand-written manuscripts aiming at hiding the content of the message
- Ciphers often contain encoded sequences of various symbols, but also *cleartext*, i.e. text written in a known language.
- Aim:
  - automatically distinguish between ciphertext and cleartext in transcribed ciphers
  - if possible, identify the language of the cleartext (often Italian, Spanish, French, German, Portuguese or Latin)
- Possible methods:
  - build and experiment with language models for historical variants of European languages
  - use existing methods for automatic language identification

Cleartext within Cipher



13017651127 4 7016011 6 2121725 0 417252 40 70148 240210136 2 701227
2202458456276 701227 21025024176 25621222405024842 52 617
13012224 2 comē la mi comāda 22225024 7012484 24 4172524 2
5072712160144246472 23847252 5602447 2 2202451224 625252

Cleartext within Cipher



cleartext

Suggestions for Projects

4. Trends in Spelling and Grammar Over Time
   - Aim:
     - developing methods for automatically identifying and analysing systematic differences in spelling and/or syntax between texts written in different time periods
   - a successful system of this kind would be very useful for e.g. historical linguists interested in language change