Standardization and orthographic variation in Late Modern Dutch witness depositions

Sara Budts, Yoshi Malaise and Rik Vosters (VUB)

CLARIAH-VL
Open Humanities Service Infrastructure

fwo

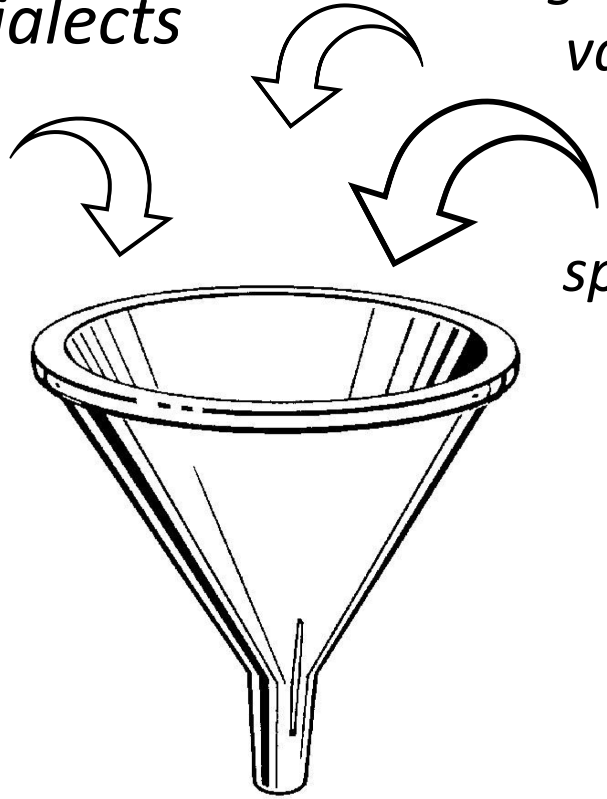
VUB B·TXT BRUSSELS DIGITAL TEXT LAB CORE FACILITY

Introduction

- Dutch language history: orthography and standardization – sociolinguistic perspectives
    - Vosters, R., & Villa, L. (Eds.). (2015). *The historical sociolinguistics of spelling*. Special issue of *Written Language & Literacy,* 18 (2).
    - Rutten, G., & Vosters, R. (Eds.). (2020). *Revisiting Haugen. Historical-sociolinguistic perspectives on standardization.* Special issue of *Language Policy* 19 (2).
- Measuring spelling variation
    - Existing methods usually based on lemmatization
    - Quantifying distance from modern standard language forms
    - Teleological bias: modern standard as a basis from which to project back into the past

Teleological bias in language history

- Tunnel / funnel view of standardization
  - Trudgill, P., & Watts, R. (2002). *Alternative histories of English*. Routledge.
  - Watts, R. (2012). Language myths. In J. M. Hernandez Campoy & J. C. Conde Silvestre (Eds.), *The handbook of historical sociolinguistics* (pp. 585–606). Wiley-Blac
- Standard language ideology
  - Milroy, J., & Milroy, L. (1985). *Authority in language. Investigating language prescription and standardisation*. Routledge and Kegan Paul.



dialects

regional variation

spelling variation

modern standard

Case and corpus (1)

- witness depositions and interrogations:
    - Bruges
    - 1709-1790 and 1838-1891
    - 3190 documents
    - 502 lawsuits
    - 1 122 785 words

- ”Witnesses” project – VUB (2018-2021)
    - Citizen science approach
    - crowd-sourced transcription through MADOC platform (https://docs.madoc.io)

Case and corpus (2)

- **Document-related** variables:
  - **Type**: deposition (75%) vs. report (23%)
  - **Scribe**: 246 individual clerks
- **Role-** and **crime-related** variables:
  - **Role**: witness (75%), suspect (14%), victim (11%)
  - **Gender** of interrogee: female (24%), male (62%)
  - **Age** of interrogee (between 5 and 88)
  - **Profession** of interrogee
  - **Crime** (e.g. theft, assault, murder, libel, …)
- Previous research
  - Serwadczak, M. (2024). (De)constructing “verbatimness”: a study of speech reporting strategies in the late modern Flemish courtroom. Journal of Historical Sociolinguistics. Online first.
  - Serwadczak (2024). Actual words were not included: Orality, literacy, and entextualization in historical criminal records (1700-1900). Vrije Universiteit Brussel, PhD dissertation.

How to find clusters of spelling variants?

1. Turn corpus into list of word pairs
2. Get language model
3. Get initial training data
4. Train and annotate (iteratively)
5. Manually check results
6. Group pairs into variant clusters

Technical information in our online material

1. Get a list of word pairs

- Spelling variants can be identified in pairwise fashion
  - take all words in vocabulary and pair them with each other
    - “howse” – “house” -> spelling variant
    - “howse” - ”noise” -> no spelling variant

- Enormous number of candidate pairs
  - 43 128 words in the Bruges part of the Testimonials corpus

    43 128 x 43 128 combinations

    = 1 860 024 384 word pairs

- Reduction of candidate list (levenshtein distance + frequency)

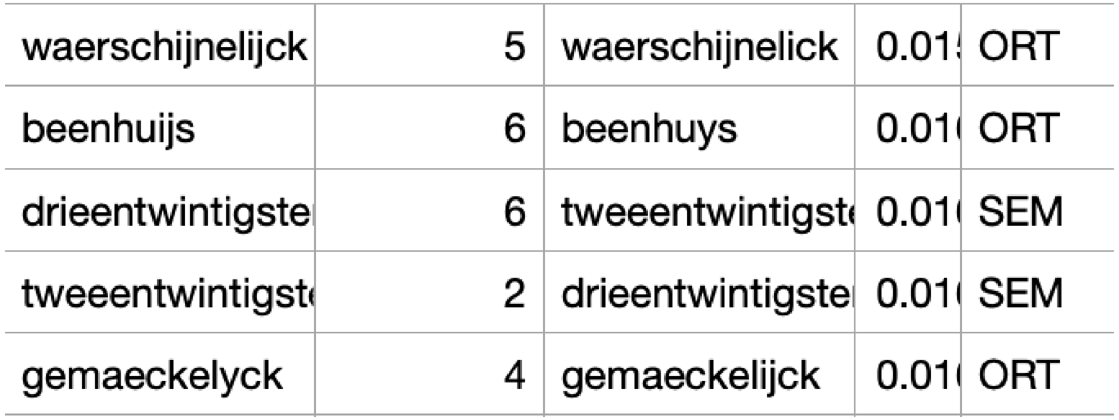
  325 794 possible pairs left

2. Get a language model

- Train a language model that predicts whether two words are spelling variants
- GysBERT (Manjavacas & Fonteyn 2022)
    - Transformer model (LLM) from BERT-series
    - Trained on Dutch historical texts (1500-1950)
        - already “gets” the meaning and grammar of historical Dutch
    - Computes with parts of words instead of full words
        - “**ge**daan vs “**ghe**daan”

3. Get initial training data

- True pairs are rare, but we need many of them to start training
- Exploit what GysBERT has already learnt about spelling variation
- Workflow:
  - Feed every word pair to GysBERT
  - Let GysBERT compute the similarity (interchangeability)
  - Sort list by decreasing similarity
  - Manually label the top 4K pairs
- Result: 4K labelled word pairs for training



waerschijnelijck	5	waerschijnelick	0.01	ORT
beenhuijs	6	beenhuys	0.01	ORT
drieentwintigste	6	tweeentwintigst	0.01	SEM
tweeentwintigst	2	drieentwintigste	0.01	SEM
gemaeckelyck	4	gemaeckelijck	0.01	ORT

4. Train your model

- Initial collection of training data
  - 4K word pairs annotated
- Iterative learning procedure: train, annotate and repeat
  - First training cycle
    - macro F1: 0.78042
    - Annotate 1K extra pairs
  - Second training cycle
    - macro F1: 0.89028
    - Annotate 1K extra pairs
  - Third training cycle
    - macro F1: 0.89748
    - little improvement, so time for evaluation
- Compute probability of being spelling variants for all unlabelled pairs

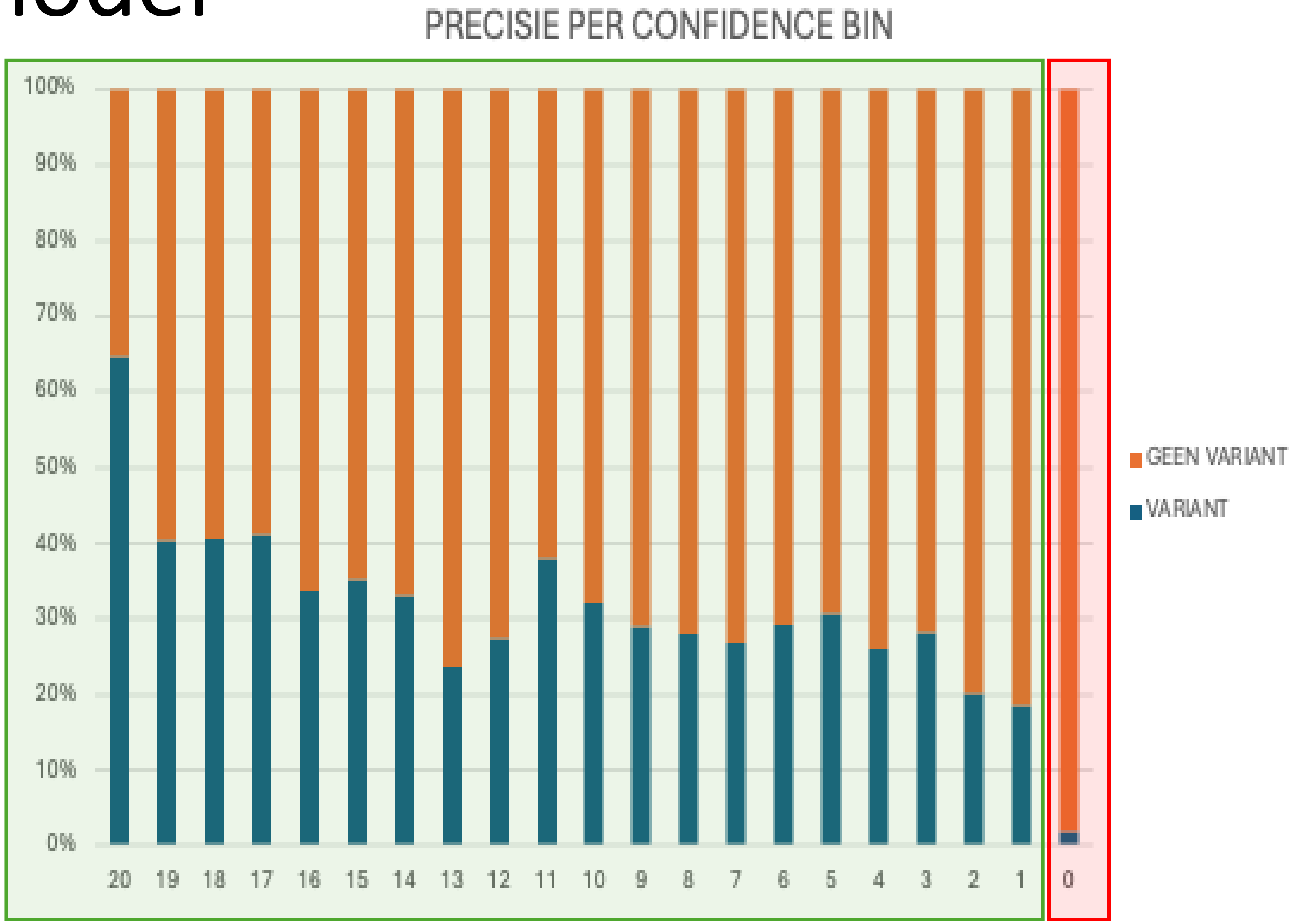
5. Evaluate your model

- Manual annotation
  - all word pairs with over 5% chance of being spelling variants

  = our "harvest"

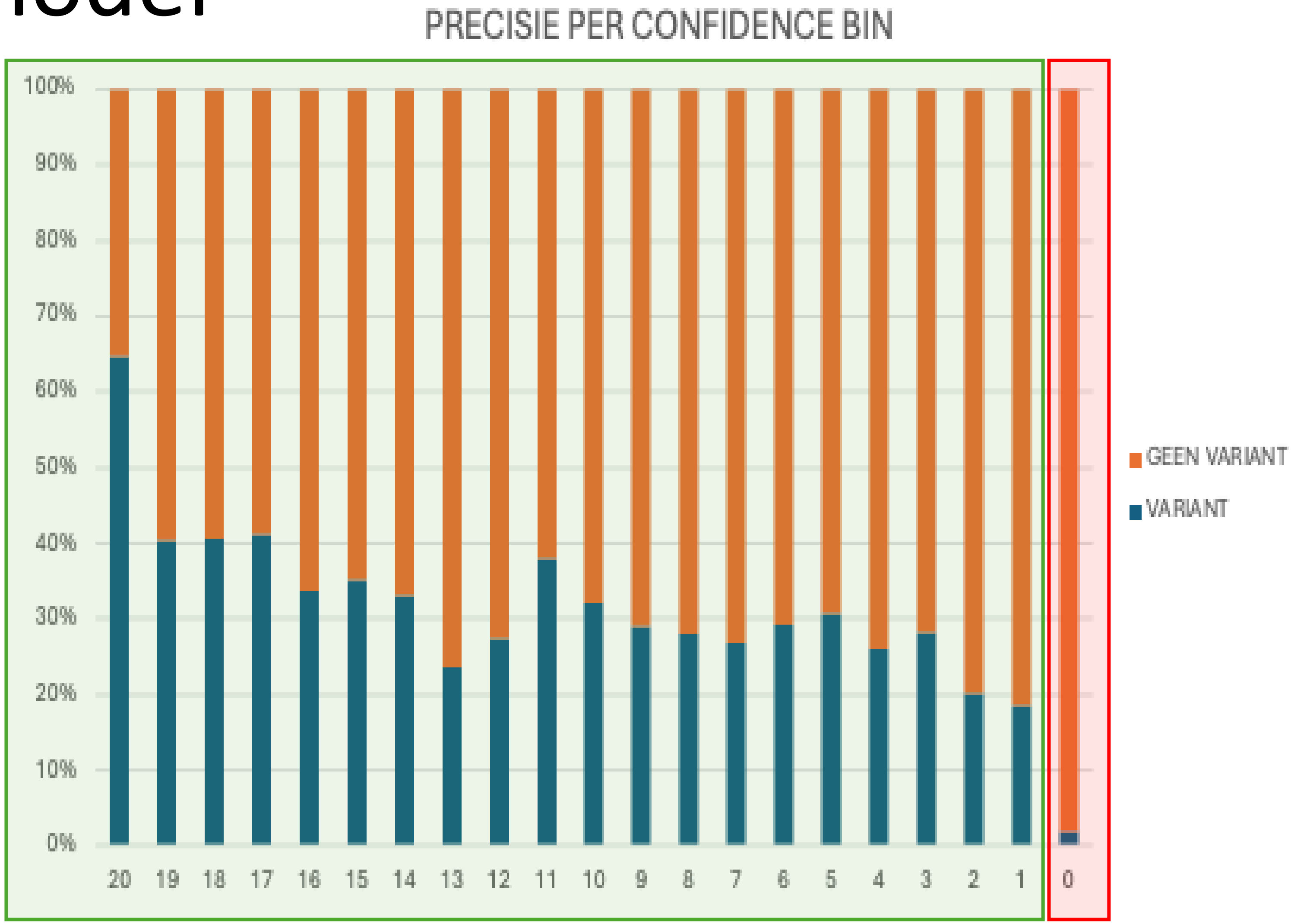
  - 500 word pairs with 0% chance

  = check what we've missed



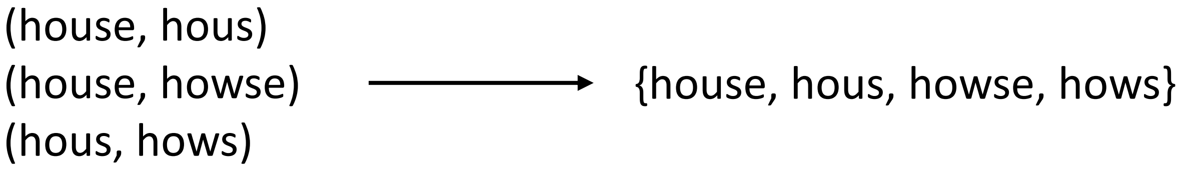
5. Evaluate your model

- Manual annotation:
  - all pairs with over 5% chance of being spelling variants
  - sample with 0% chance
    - 3% error rate
    - BUT 309 280 pairs
    - ca. 7072 variants not found
- Only 34% of variation found



6. Group pairs into variant clusters

- From pairs back to clusters



(house, hous)
(house, howse)
(hous, hows)

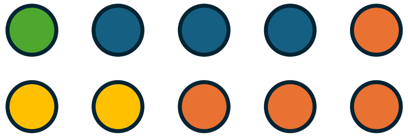
{house, hous, howse, hows}

- Example of cluster:

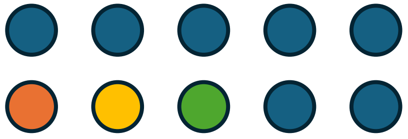
ghelijck, gelyck, gelyke, ghelyck, gelijcke, gelÿck, gelyk, ghelick, ghelÿck, gelijck, gelijke, gelijk, gelick, gelÿk, gelycke, gelijken, gelyken

Variation metric

- Relative frequency of most frequent word type within cluster (max_rel)
  - e.g. 10 tokens in total, 8 of which are the same type -> 0.8
  - To measure variation: 1 – max_rel
  - Hypothesis: variation decreases



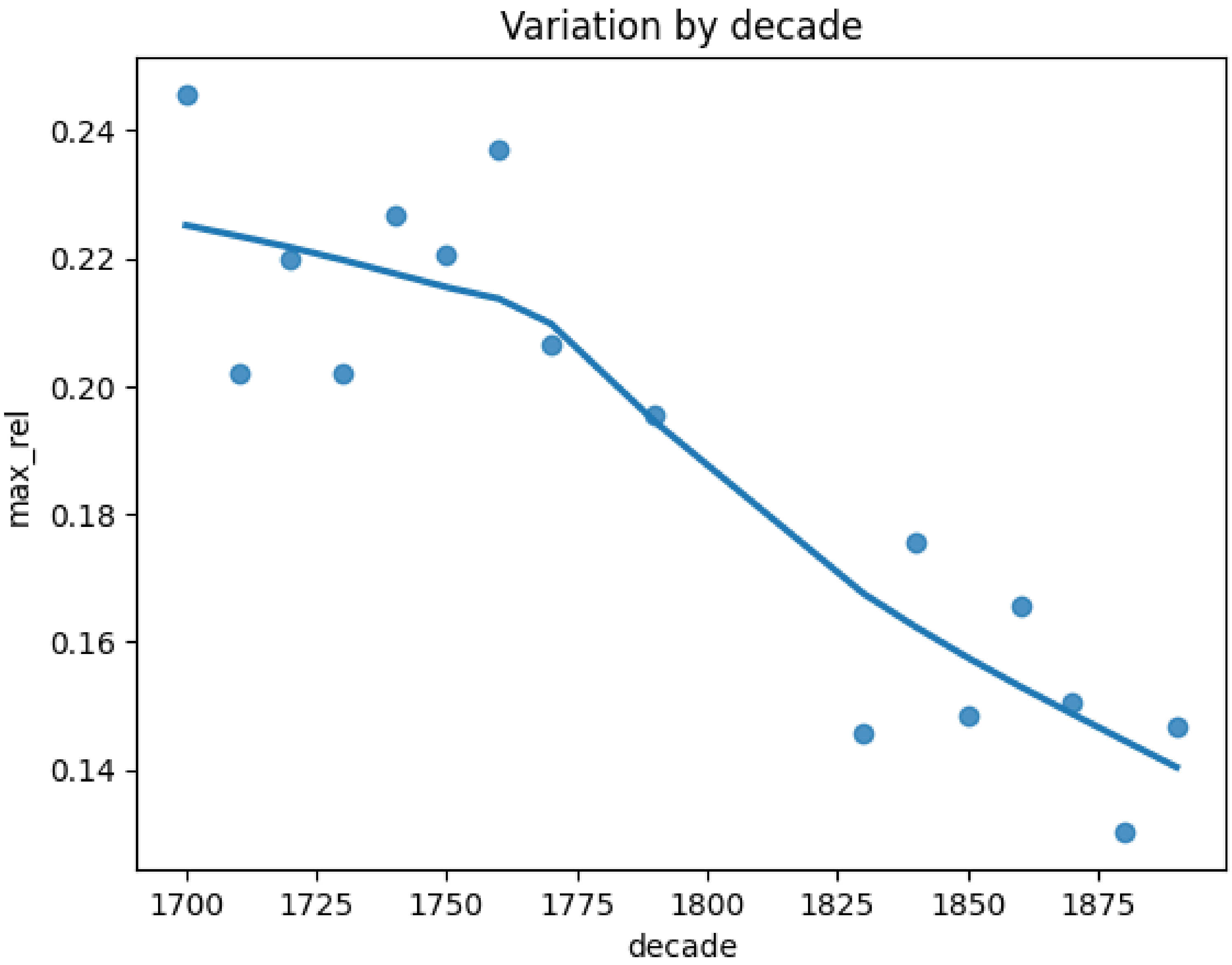
max_rel = 4/10
1- max_rel = 6/10
variation = 60%

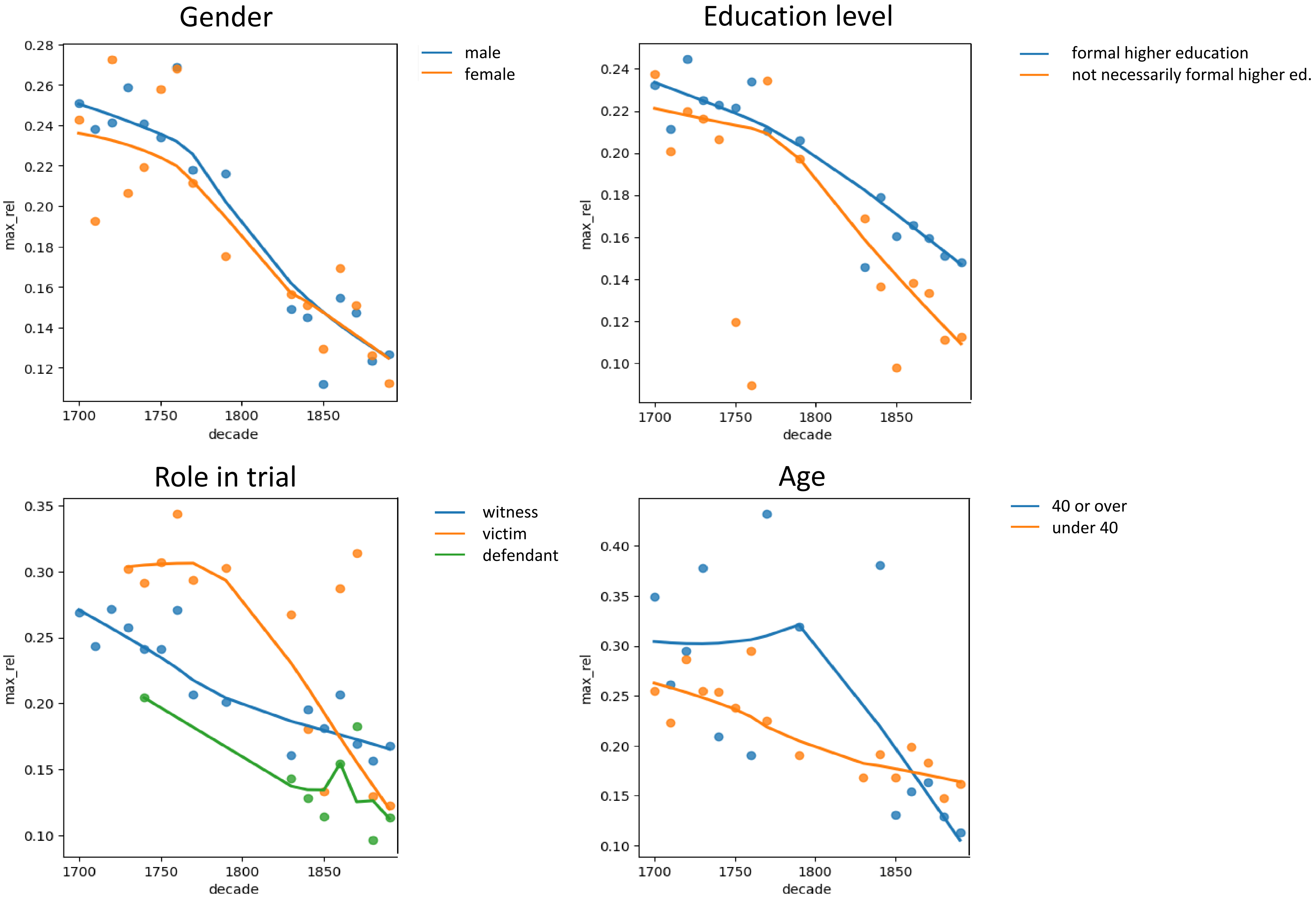


max_rel = 7/10
1- max_rel = 3/10
variation = 30%

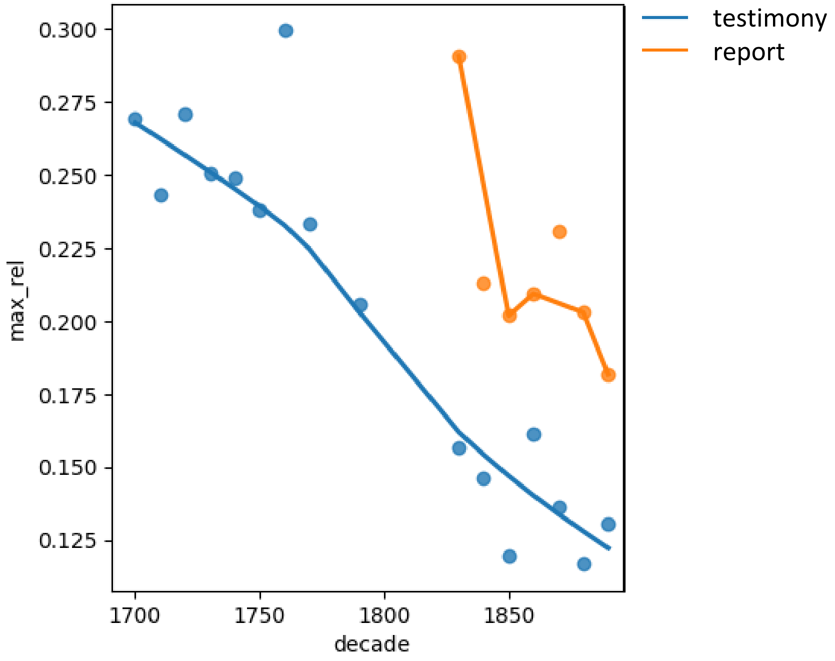
Sociolinguistic analysis

1. Did spelling variation decrease over time?
2. Does spelling variation correlate with role-related variables
3. Does spelling variation correlate with document-related variables?
4. Do individual scribes spell systematically, or do they spell the same word in various ways too?

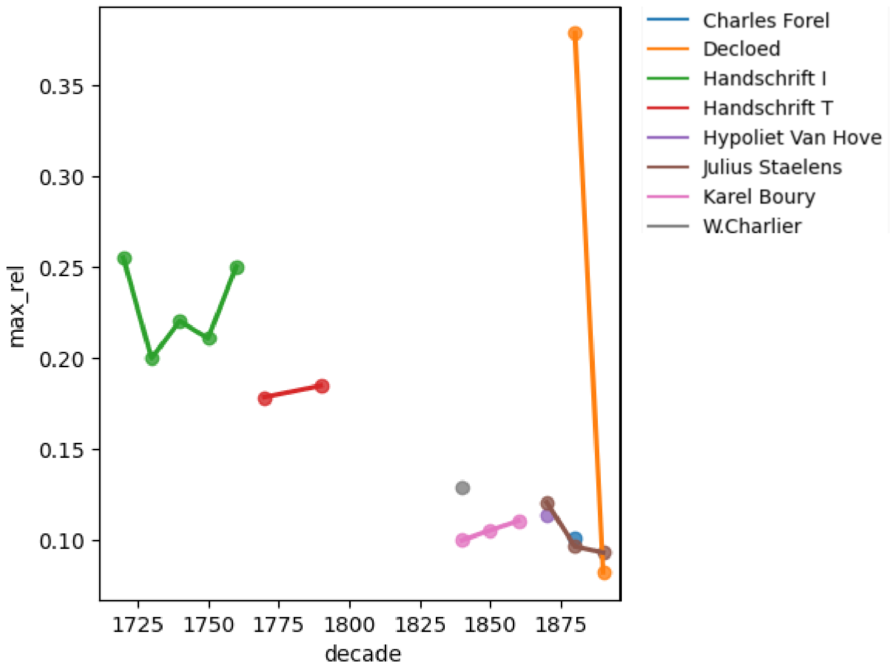

Variation by decade
max_rel
0.24
0.22
0.20
0.18
0.16
0.14
1700
1725
1750
1775
1800
1825
1850
1875
decade


Gender
0.28
0.26
0.24
0.22
0.20
0.18
0.16
0.14
0.12
max_rel
1700
1750
1800
1850
decade
male
female
Education level
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0.10
max_rel
1700
1750
1800
1850
decade
formal higher education
not necessarily formal higher ed.
Role in trial
0.35
0.30
0.25
0.20
0.15
0.10
max_rel
1700
1750
1800
1850
decade
witness
victim
defendant
Age
0.40
0.35
0.30
0.25
0.20
0.15
0.10
max_rel
1700
1750
1800
1850
decade
40 or over
under 40

Document type



Scribe



Conclusions (case study)

- Document variables > role variables for spelling variation

  = social profile of scribes matters more than that of witnesses

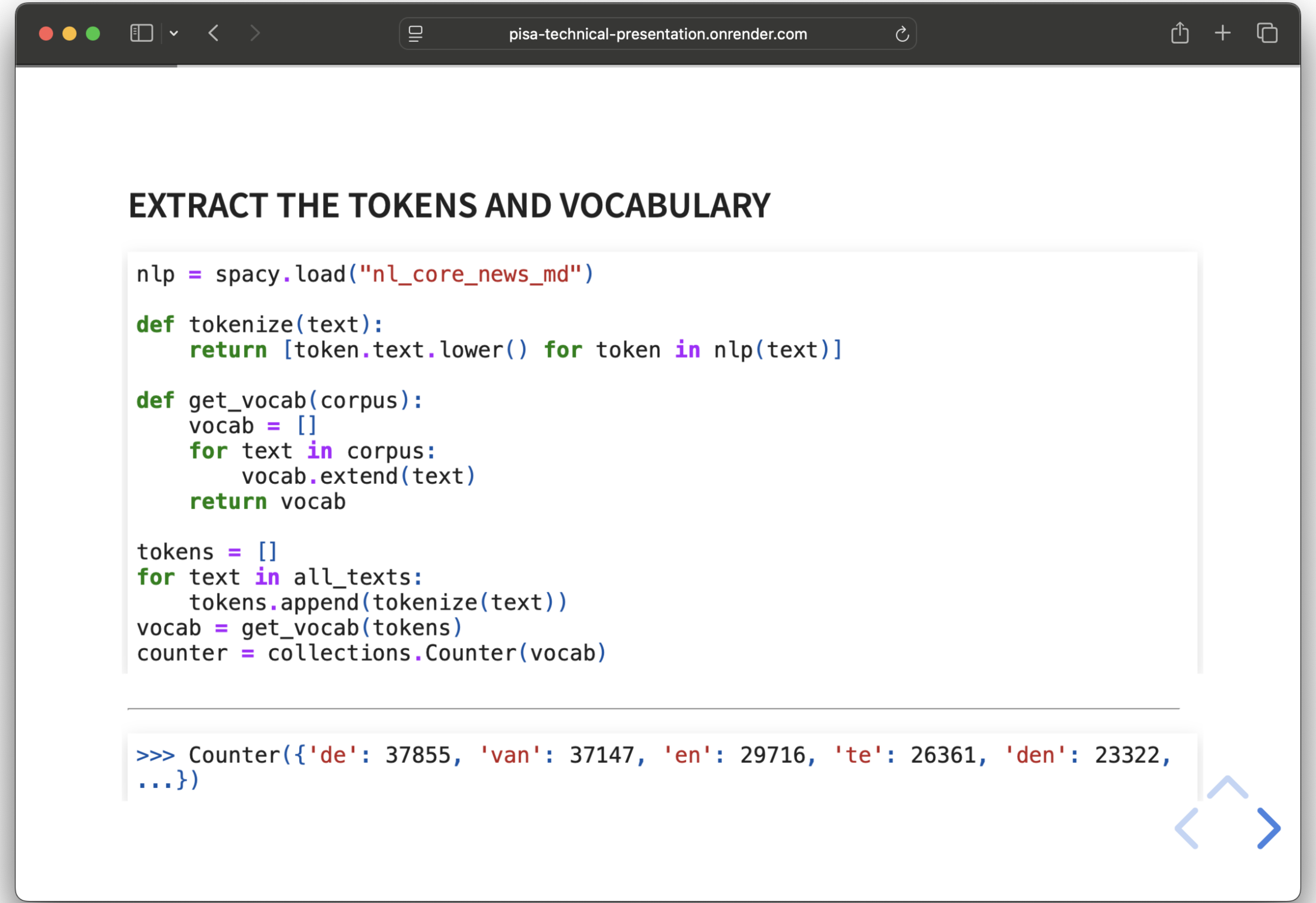
  <-> morphological and syntactic variation

  - because spelling cannot be transmitted (?)

- Spelling of different scribes fluctuates in line with what’s common for their time frame

Conclusions (methodology)

- It’s a recipe: very hands-on
- Limitations
  - Only 34% of actual spelling variation is included
  - Inclusion/Exclusion is not random
- Strengths
  - Bottom-up, non-teleological analysis of spelling variation
  - Possibility to find things that you weren’t already expecting
  - Can be applied to other languages
- **Online companion and tutorial**: https://btxt.research.vub.be/en/resources



EXTRACT THE TOKENS AND VOCABULARY

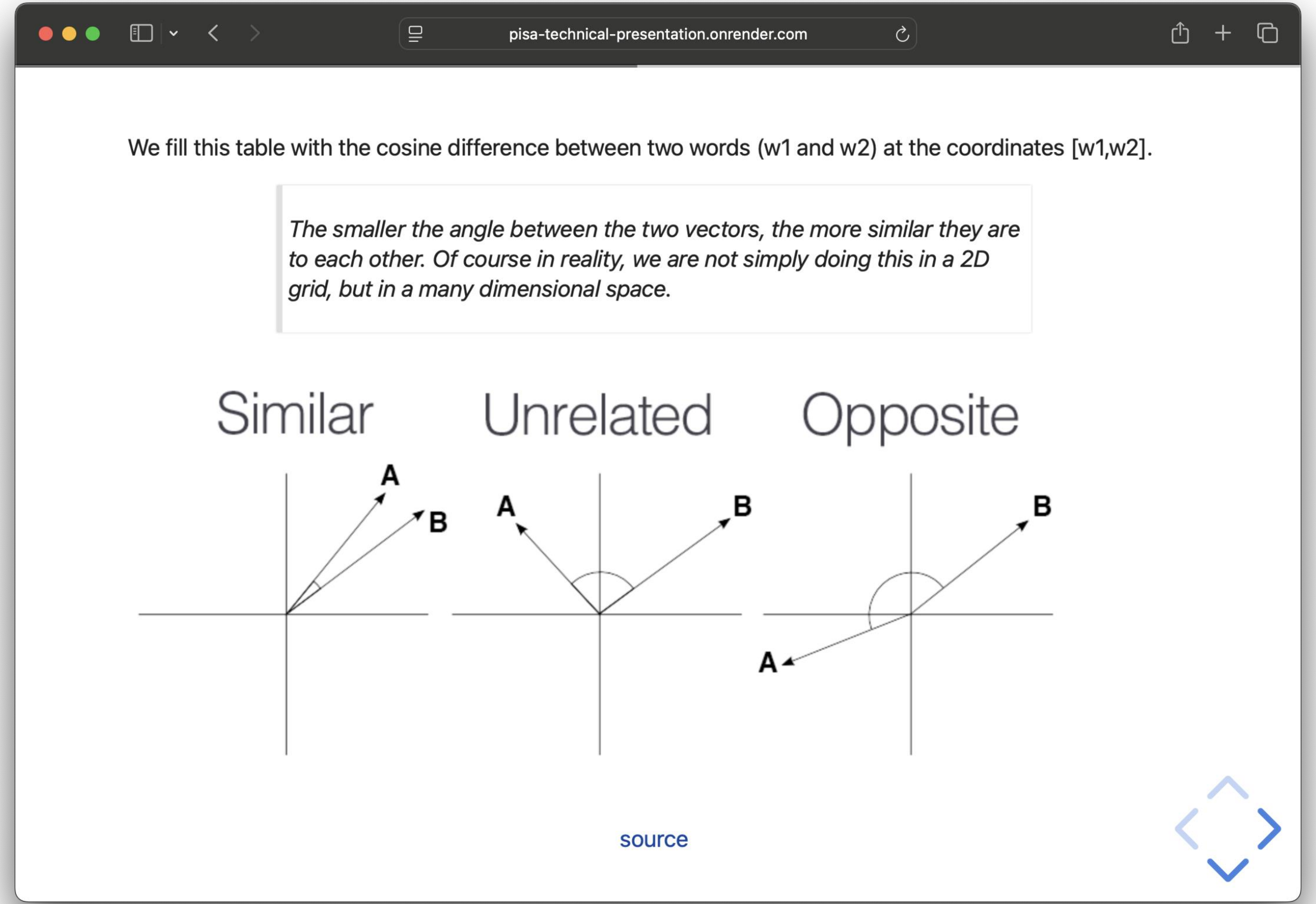
```python
nlp = spacy.load("nl_core_news_md")

def tokenize(text):
    return [token.text.lower() for token in nlp(text)]

def get_vocab(corpus):
    vocab = []
    for text in corpus:
        vocab.extend(text)
    return vocab

tokens = []
for text in all_texts:
    tokens.append(tokenize(text))
vocab = get_vocab(tokens)
counter = collections.Counter(vocab)
```

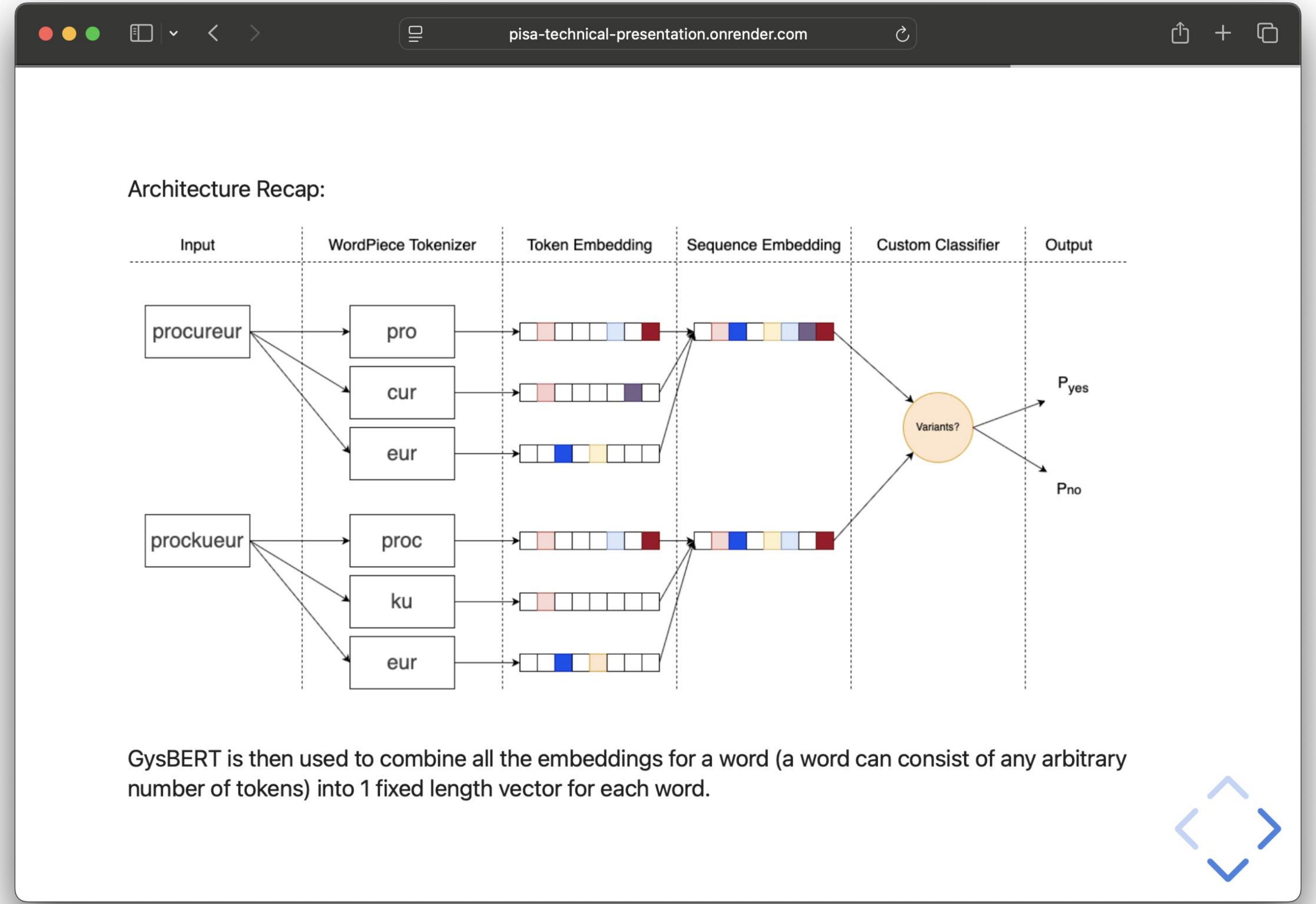
```
>>> Counter({'de': 37855, 'van': 37147, 'en': 29716, 'te': 26361, 'den': 23322,
...})
```



We fill this table with the cosine difference between two words (w1 and w2) at the coordinates [w1,w2].

> *The smaller the angle between the two vectors, the more similar they are to each other. Of course in reality, we are not simply doing this in a 2D grid, but in a many dimensional space.*

source



Architecture Recap:

GysBERT is then used to combine all the embeddings for a word (a word can consist of any arbitrary number of tokens) into 1 fixed length vector for each word.