CLARIAH Chaining Search: A Platform for Combined Exploitation of Multiple Linguistic Resources

Peter Dekker, Mathieu Fannee, Jesse de Does
Dutch Language Institute (INT)

October 1, 2019

CLARIN Annual Conference 2019
Leipzig, Germany

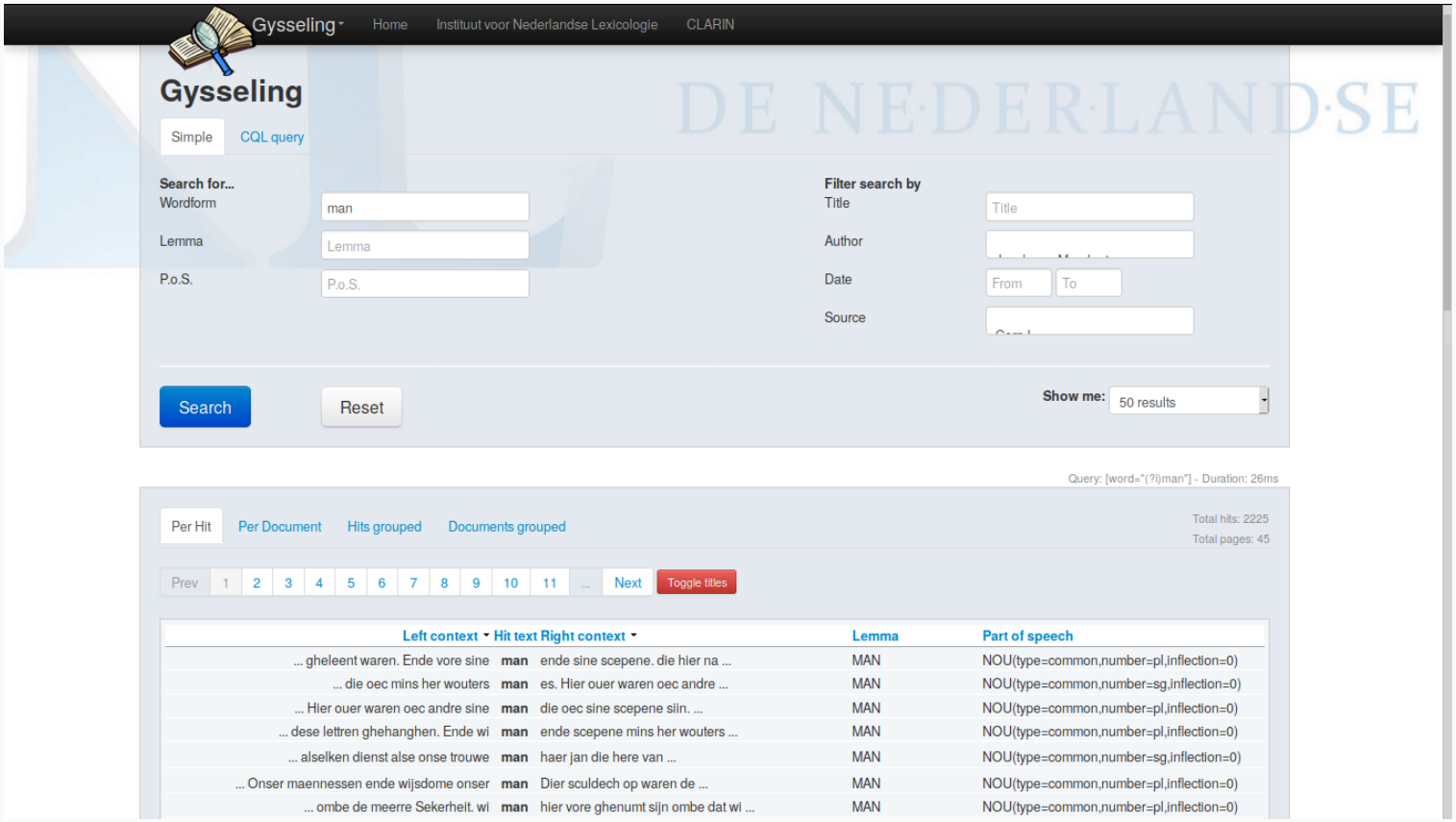
Why CLARIAH Chaining search?

- Perspective: linguist (with some computational knowledge)
- Goal: combine heterogeneous CLARIN resources: corpora, lexica, treebanks
- Current flawed options
  - Access web interfaces
  - Download entire dataset

Current option: Web interface

/instituut voor de Nederlandse taal/

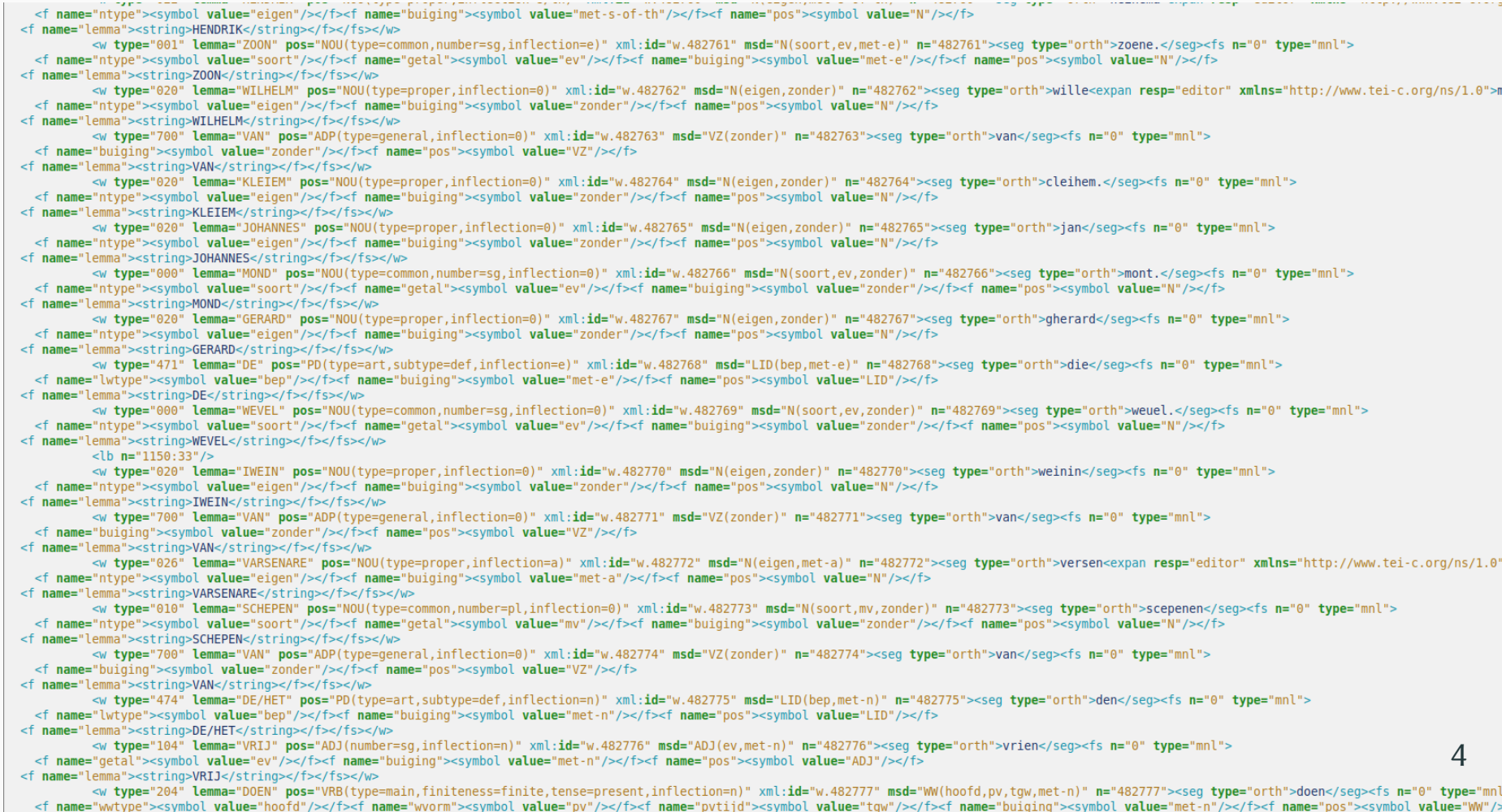
- Corpus Gysseling: 13th century Dutch
- Web interface:
  `http://gysseling.corpus.taalbanknederlands.inl.nl/gysseling/page/search`



Current option: Download entire dataset

/instituut voor de Nederlandse taal/

- Corpus Gysseling: 13th century Dutch
- Download dataset: `https://ivdnt.org/taalmaterialen/102-taalmaterialen/2014-tstc-corpus-gysselingh`



CLARIAH searchability

- CLARIAH: digital research infrastructure for arts and humanities in The Netherlands
- Levels of searchability:
  - Local search: resource available as web service
  - Federated search: resources of same type queried as single resource
  - Chaining search: heterogeneous sources combined, sequential search workflows

CLARIAH chaining search

A Python library and Jupyter web interface to easily combine exploration of linguistic resources published in the CLARIN/CLARIAH infrastructure, such as corpora, lexica and treebanks.

Implementation: earlier idea

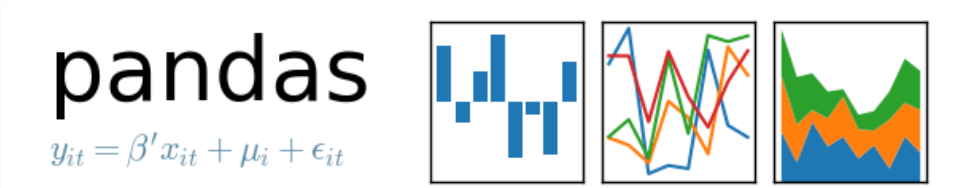
/instituut voor de Nederlandse taal/

- Large SPARQL query to query multiple resources
- But:
  - Not all resources available as Linked Open Data, queried with SPARQL
  - Query becomes highly complex

```
 values ?n_ontolex_writtenRep  { "{{word}}" } } .
?n_entry a ontolex:LexicalEntry .
?n_entry  ud:pos ?udobj . ?udobj rdfs:label ?udpos .
?n_form a ontolex:Form .
?n_sense a ontolex:LexicalSense .
?n_syndef a diamant:SynonymDefinition .
?n_sensedef a lemon:SenseDefinition .
?n_syndef diamant:definitionText ?n_syndef_definitionText  .
?n_sensedef diamant:definitionText ?n_sensedef_definitionText  .
?n_entry ontolex:canonicalForm ?n_form  .
?n_entry ontolex:sense ?n_sense  .
?n_sense lemon:definition ?n_syndef  .
?n_sense lemon:definition ?n_sensedef  .
 ?n_sense diamant:attestation ?n_attest_show .
 ?n_sense diamant:attestation ?n_attest_filter .
 ?n_attest_show diamant:text ?n_q_show .
 ?n_attest_filter diamant:text ?n_q_filter .
 ?n_attest_show a diamant:Attestation .
 ?n_attest_filter a diamant:Attestation .
 ?n_q_filter a diamant:Quotation .
 ?n_q_show a diamant:Quotation .
 ?n_q_filter diamant:witnessYearFrom ?wy_f_filter .
 ?n_q_filter diamant:witnessYearTo ?wy_t_filter .
 ?n_q_show diamant:witnessYearFrom ?wy_f_show .
 ?n_q_show diamant:witnessYearTo ?wy_t_show .
 FILTER (xsd:integer(?wy_f_show) >= {{timelineStart}})
 FILTER (xsd:integer(?wy_t_show) >= {{timelineStart}})
 FILTER (xsd:integer(?wy_f_show) <= {{timelineEnd}})
 FILTER (xsd:integer(?wy_t_show) <= {{timelineEnd}})
bind("searchLemma" as ?resultType)  .
} UNION
```

Implementation: now

- Python library
  - Based on pandas DataFrames McKinney (2011)
  - Four modules: `search`, `process`, `ui` and `utils`
- Jupyter notebooks
  - Examples notebook
  - Sandbox: start coding yourself
  - Sailing letters case study



Resources: Supported protocols

- Corpora
  - CLARIN Federated Content Search endpoint (Stehouwer et al., 2012)
    - Queried via CQL
    - More than 1200 freely available corpora
  - BlackLab corpus search engine (De Does et al., 2017)
    - Queried via CQL
    - More search options, e.g. metadata filtering
- Lexica
  - Linked Open Data, *Ontolex* lexicon model (McCrae et al., 2017)
    - Queried via SPARQL
  - Internal INT *lexicon service* API
- Treebanks
  - CGN (corpus of spoken Dutch) and Lassy (written Dutch)
    - Queried via XPath query

Library: search

Search in linguistic resource:

```
results = create_lexicon(lexicon_name).lemma(
    word).search()
results = create_corpus(corpus_name).pattern(
    pattern).search()
results = create_treebank(treebank_name).
    pattern(pattern).search()
```

Create results table (Pandas DataFrame) with keywords-in-context:

```
df = results.kwic()
```

Library: `utils`

`utils` provides functions general operations applied to search results tables.

```
diff = column_difference( df1["lemma_0
    "], df2["lemma_0"] )
```

Library: process

`process` handles linguistic processing of data.

```
df_lexicon = extract_lexicon(df_corpus)
```

Library: `ui`

Operations regarding showing the user interface and loading/saving data

```
save_dataframe(df, "test.csv")
df = load_dataframe("test.csv")
```

Chaining in practice: adjectives without -e

Look for adjectives which should have ending -e, but miss it with determiner *een*.

First, search in corpus.

```
df_corp = create_corpus("openchn").
    pattern('[pos="DET"&lemma="een"][
    word=".*[^e]$" & pos="AA.*degree=
    pos.*"][pos="NOU.*gender=[fm].*"]')
    .search().kwic()
```

Chaining in practice: adjectives without -e

Search in lexicon.

```
df_lex = create_lexicon("molex").lemma
    ('(.+)[^e]$').pos('ADJ(degree=pos)'
    ).search().kwic()
final_e_condition = df_filter(df_lex["
    wordform"], 'e$')
df_lexicon_form_e = df_lex[
    final_e_condition ]
```

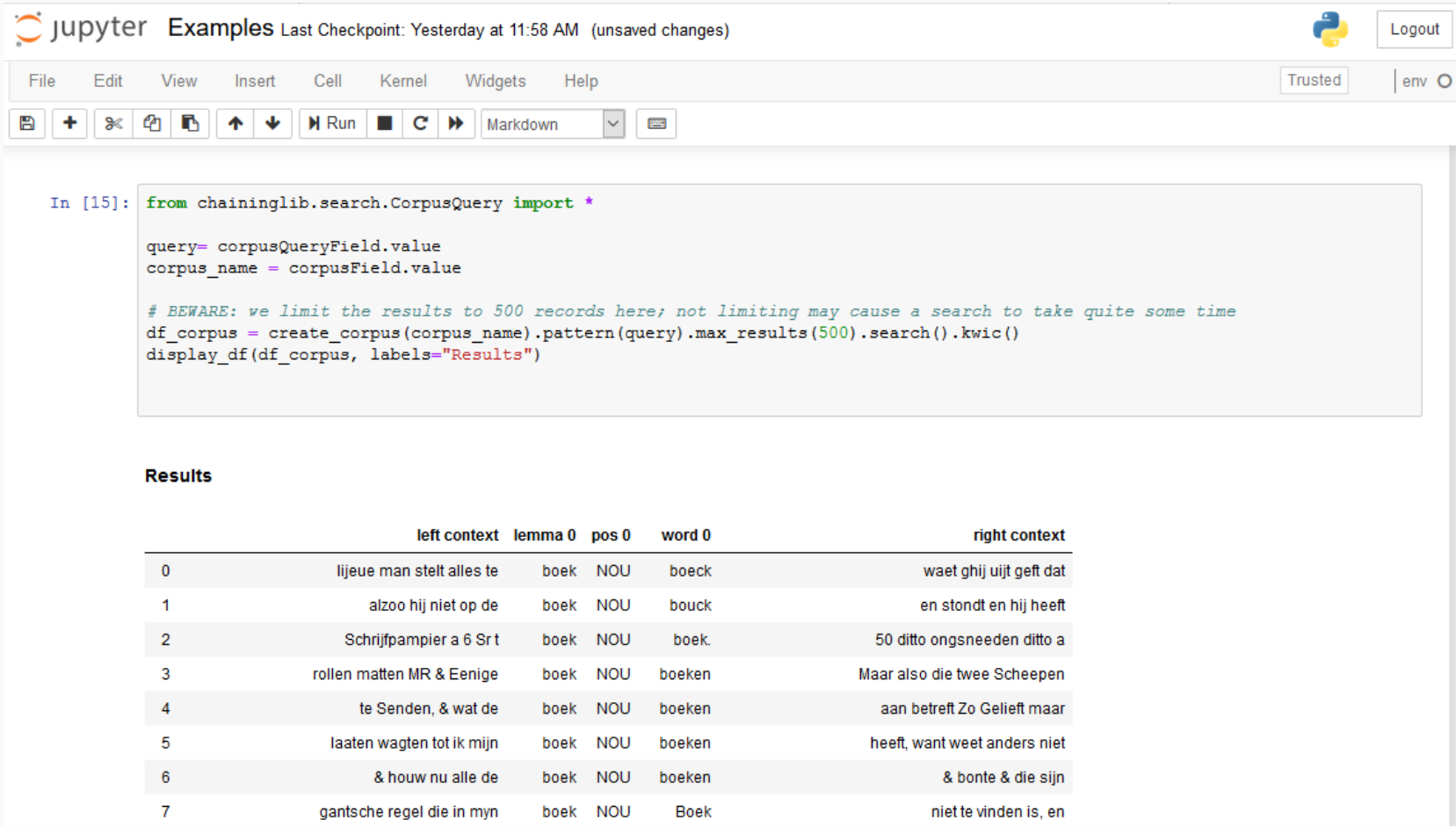
Chaining in practice: adjectives without -e

Filter corpus using results from lexicon.

```
e_forms = set(df_lexicon_form_e.lemma)
cond = df_filter(df_corp["word 1"],
    pattern=e_forms, method="isin")
result_df = df_corp[ cond ]
```

Jupyter notebook

/instituut voor de Nederlandse taal/



Results

	left context	lemma 0	pos 0	word 0	right context
0	lijeue man stelt alles te	boek	NOU	boeck	waet ghij uijt geft dat
1	alzoo hij niet op de	boek	NOU	bouck	en stondt en hij heeft
2	Schrijfpampier a 6 Sr t	boek	NOU	boek.	50 ditto ongsneeden ditto a
3	rollen matten MR & Eenige	boek	NOU	boeken	Maar also die twee Scheepen
4	te Senden, & wat de	boek	NOU	boeken	aan betreft Zo Gelieft maar
5	laaten wagten tot ik mijn	boek	NOU	boeken	heeft, want weet anders niet
6	& houw nu alle de	boek	NOU	boeken	& bonte & die sijn
7	gantsche regel die in myn	boek	NOU	Boek	niet te vinden is, en

Case study: Social class and gender in sailing letters

Letters as Loot corpus (Van der Wal et al., 2012): 17th and 18th century letters from Dutch sailors, annotated with metadata

- Research question: *Which vocabulary is specific for:*
  - *Social class* (low or high)
  - *Gender* (male or female)
  - *Time period* (17th or 18th century)
- Use CLARIAH chaining search to:
  1. Retrieve data from corpus
  2. Filter and split data on metadata (class/gender/era)
  3. Compute relative frequencies of words per metadata category
  4. Compute differences (ratios) between relative frequencies
- Notebook Case_study_paper.ipynb in GitHub repository

Time period: 17th vs 18th century (high class)

Lemmata most specific for time period (highest diff in rel frequency)

17th century			
lemma	relative frequency		
	17th	18th	diff
huisvrouw	0.016	0.002	1.013
vriend	0.023	0.011	1.012
man	0.020	0.010	1.010
gezondheid	0.016	0.007	1.010
goedenacht	0.009	0.000	1.009
schipper	0.009	0.000	1.009
brief	0.022	0.014	1.008
suiker	0.010	0.002	1.008
monsieur	0.008	0.001	1.006
schip	0.023	0.017	1.006

18th century			
lemma	relative frequency		
	17th	18th	diff
heer	0.006	0.021	0.985
mijnheer	0.009	0.018	0.991
edele	0.000	0.008	0.992
jaar	0.005	0.012	0.993
mejuffrouw	0.001	0.006	0.995
kapitein	0.012	0.017	0.995
zuster	0.008	0.013	0.995
achting	0.000	0.005	0.995
familie	0.002	0.006	0.996
liefde	0.001	0.006	0.996

Social class: low vs high (17th century)

Lemmata most specific for social class (highest diff in rel frequency)

low

lemma	relative frequency		
	low	high	diff
goedenacht	0.026	0.009	1.017
hart	0.027	0.012	1.015
man	0.033	0.020	1.013
brief	0.034	0.022	1.012
Heer	0.022	0.011	1.011
gezondheid	0.026	0.016	1.009
zuster	0.016	0.008	1.008
huisvrouw	0.024	0.016	1.008
kind	0.020	0.013	1.007
zoon	0.016	0.011	1.005

high

lemma	relative frequency		
	low	high	diff
kapitein	0.005	0.012	0.993
monsieur	0.000	0.008	0.993
suiker	0.002	0.010	0.993
sinjeur	0.001	0.006	0.995
mijnheer	0.004	0.009	0.995
schipper	0.005	0.009	0.996
heer	0.002	0.006	0.996
pond	0.000	0.004	0.997
december	0.002	0.005	0.997
rekening	0.001	0.004	0.997

Gender: male vs female (17th century, low class)

Lemmata most specific for gender (highest diff in rel frequency)

male				female			
lemma	relative frequency			lemma	relative frequency		
	male	female	diff		male	female	diff
suiker	0.002	0.000	1.002	man	0.033	0.041	0.992
schip	0.024	0.022	1.002	brief	0.034	0.038	0.996
oom	0.003	0.002	1.001	hart	0.027	0.031	0.996
vriend	0.022	0.021	1.001	Heer	0.022	0.023	0.998
december	0.002	0.001	1.001	zoon	0.016	0.018	0.998
maand	0.003	0.002	1.001	allerliefste	0.005	0.006	0.998
huis	0.004	0.003	1.001	mens	0.005	0.006	0.998
port	0.002	0.001	1.001	gezondheid	0.026	0.027	0.999
vaderland	0.002	0.001	1.001	huisvrouw	0.024	0.025	0.999
stuk	0.001	0.000	1.001	reis	0.010	0.011	0.999

Gender: male vs female (17th century, high class)

Lemmata most specific for gender (highest diff in rel frequency)

male				female			
lemma	relative frequency			lemma	relative frequency		
	male	female	diff		male	female	diff
suiker	0.010	0.001	1.009	man	0.020	0.046	0.975
monsieur	0.008	0.001	1.006	brief	0.022	0.034	0.988
vriend	0.023	0.019	1.005	kind	0.013	0.022	0.991
sinjeur	0.006	0.002	1.004	zoon	0.011	0.018	0.993
december	0.005	0.001	1.004	goedenacht	0.009	0.016	0.993
goed	0.011	0.007	1.004	zuster	0.008	0.015	0.994
vracht	0.003	0.000	1.003	gezondheid	0.016	0.022	0.994
mr.	0.003	0.000	1.003	dochter	0.004	0.010	0.995
cargasoen	0.003	0.000	1.003	genade	0.008	0.013	0.995
dienaar	0.002	0.000	1.00	hart	0.012	0.017	0.995

Conclusion

- Case study shows difference in vocabulary across sociolinguistic variables, especially for gender in higher social class
- Case study proves: CLARIAH Chaining search facilitates and accelerates linguistic research
- Discussion
  - Minimum of programming knowledge is needed
  - Long computations: invoke library directly, without notebook
  - Future work: optimization for very large data sets

Interested?

- Demonstration during poster session
- Try it yourself: `https://github.com/INL/chaining-search`
  - Local install
  - Cloud instance on Azure
- FUll API reference in documentation: `https://chaining-search.readthedocs.io/en/latest/`

References

De Does, J., Niestadt, J., and Depuydt, K. (2017). Creating Research Environments with BlackLab. In Utrecht University, NL and Odijk, J., editors, *CLARIN in the Low Countries*, pages 245–257. Ubiquity Press.

McCrae, J. P., Bosque-Gil, J., Gracia, J., Buitelaar, P., and Cimiano, P. (2017). The ontolex-lemon model: development and applications. In *Proceedings of eLex 2017 conference*, pages 19–21.

McKinney, W. (2011). pandas: a Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing*, 14:9.

Stehouwer, H., Durco, M., Auer, E., and Broeder, D. (2012). Federated Search: Towards a Common Search Infrastructure. *LREC 2012*, page 5.

Van der Wal, M. J., Rutten, G., and Simons, T. (2012). Letters as loot: Confiscated Letters filling major gaps in the History of Dutch. In Dossena, M. and Del Lungo Camiciotti, G., editors, *Pragmatics & Beyond New Series*, volume 218, pages 139–162. John Benjamins Publishing Company, Amsterdam.