heidelgram.uni-heidelberg.de

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

The Case for Custom Software Development

HGSimpleCorpusNetwork – A Network Analysis Toolbox for (Historical) Corpora

Beatrix Busse @BeatrixBusse / Ingo Kleiber @KleiberIngo

Gefördert durch DFG Deutsche Forschungsgemeinschaft

HeidelGram Project

The *HeidelGram* project has **two interrelated objectives**:

- Compiling and analyzing a corpus of historical English grammar books (1550 – 1900)
- Combining corpus-based historical linguistics and the analysis of networks

(cf. Busse, Gather, Kleiber 2018, 2019, forth; Busse and Kleiber 2018)

So far:

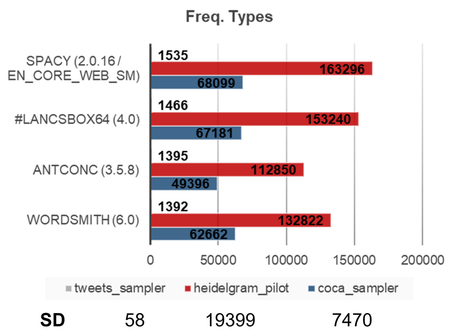
- Pilot-corpus of 19th-century grammar books (approx. 3m words)
- Analysis of the references made to grammarians/grammars in these grammar books
- Identification of generalizable Verbal Hygiene (Cameron [1995] 2012) patterns
- Development of a specialized **software toolkit for the project**

The Role of Software in Linguistic Research

"The functionality offered by software tools largely dictates what corpus linguistics research methods are available to a researcher." **and** "… differences in the way tools are designed will have an impact on almost all corpus analyses." (Anthony 2013: 141, 151)

"[Relying on existing programs] not only severely **limits the possibilities for exploring corpus data**, but also **introduces unknown factors** into the research, as it is not always obvious how the software will handle certain features of language." (Mason 2008: 155)

→ **We need to consider software as a key part of research which is as important as theory, methodology, and data!**



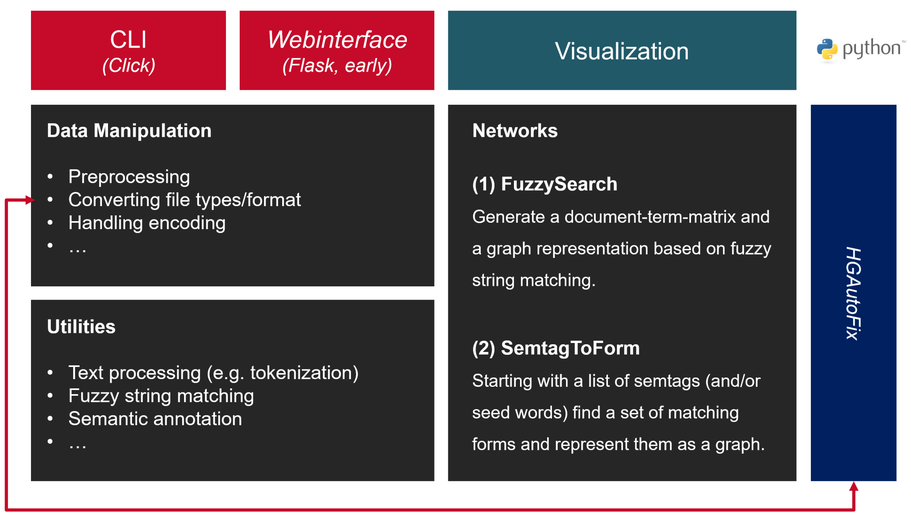
Requirements for Academic Software

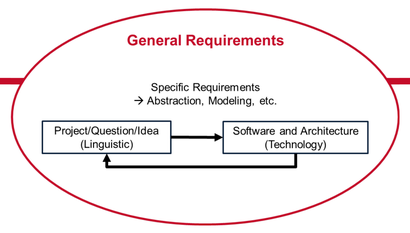
- Reproducibility
- Availability (Will the software be available in 5/10/20 years?)
- Transparency (e.g. models, measures, pre-/post-processing)
- Flexibility (cf. Mason 2000: 4)
- Compatibility (e.g. standard and open formats)
- "Pipelinability" (Will the software work as part of a data-analysis pipeline?)
- Usability + Sensible defaults
- Openness (Will there be licensing issues?)
- …

Project-Specific Software

- We don't want to be limited by existing tools and approaches
- We want to be able to solve complex tasks highly specific to our research
- We want to be able to fully understand the tools and the underlying models

HGSimpleCorpusNetwork – Requirement-Driven Agile Development





General Requirements: The toolbox needs to …

1. grow and be adapted as the project grows
2. be able to handle textual data from a variety of periods (Historical!)
3. be able to handle (and mitigate) uncleaned OCR data
4. be able to work as part of an automized analysis pipeline
5. be able to interface with network/graph analysis toolkits such as *Gephi*

Specific Required Functionalities (Example) → Lead to: **FuzzySearch**

Generate document-term matrices based on sets of search terms and OCR-based corpora.

Reason: We want to analyze which grammarians (and grammars) are referenced where (by whom) in a corpus of grammar books. (cf. Busse, Gather, Kleiber 2019)

Challenges: (1) Uncleaned OCR data is not well-structured and contains spelling errors. (2) The mere existence of a token (e.g. *Crombie*) does not necessarily entail a reference.

Lessons Learned *Don't develop for the sake of developing!*

- Software decay/entropy → constant refactoring (expensive, time-consuming)
- External dependencies (e.g. APIs) are a risk → e.g. *UcrelSemTaggerSoapService* going down
- Don't reinvent the wheel, especially if good models and approaches are available (e.g. tokenization, string matching)
- **Modeling parts of the theory in software leads to new insights into both theory and analysis**
- Developing project-specific software forces you to have extremely good knowledge of your data and methodology
- Rapidly prototyping new (computational) approaches to analysis fosters creativity (this requires you to have a modular framework and good pipelines)

Best Practices

- **Open Source Software** (OSS) + **DOI** + **Archive repository**
- Strong **version control** (e.g. Git) + branching → documentation of the process
- Testing / **Test-Driven Development** (Unit Tests, Integration Tests, Regression Tests)
- Run **simulations** and test against (reasonable) ranges and means → testing fuzzy returns
- **Continuous integration** + **Sanity/coverage** checks before pushing to remote
- Good (and complete) **documentation** (instructions, examples, self-documenting code, automated API documentation)
- **Reproducible runtimes** (e.g. *Docker*)
- **Limited reliance** on external (and specifically proprietary) modules/libraries

Works Cited

- **Anthony**, Laurence. 2013. "A Critical Look at Software Tools in Corpus Linguistics." *Linguistic Research* 30 (2): 141–61.
- **Cameron**, Deborah. (1995) 2012. *Verbal Hygiene.* The politics of language. London: Routledge.
- **Harpole**, Alice. 2017. "Sustainable Scientific Software Development." europython 2017, Rimini, July 12. https://ep2017.europython.eu/conference/talks/sustainable-scientific-software-development.
- **Mason**, Oliver. 2000. *Programming and Corpus Linguistics.* Edinburgh: Edinburgh University Press.
- **Mason**, Oliver. 2008. "Developing Software for Corpus Research." *International Journal of English Studies* 8 (1): 141–56.
- **Murray**, Gerald. 1847. *The Reformed Grammar or Philosophical Test of English Composition: Written for the Assistance of Teachers and Satisfaction of Learners.* London: Darton and Co. Holborn Hill.
- **Busse**, Beatrix, and Ingo **Kleiber**. 2018. "A Network of Evaluative Terms in 19th-Century British Grammars: Methodological Challenges and Practical Solutions." ICAME 39, Tampere, Finnland, 2018.
- **Busse**, Beatrix, Kirsten **Gather**, and Ingo **Kleiber**. 2018. "Assessing the Connections between English Grammarians of the Nineteenth Century: A Corpus-Based Network Analysis." In *Grammar and Corpora 2016*, edited by Eric Fuß, Marek Konopka, Beata Trawiński, and Ulrich H. Waßner, 435–42. Heidelberg: Heidelberg University Publishing.
- **Busse**, Beatrix, Ingo **Kleiber**, and Kirsten **Gather**. 2019. "Paradigm Shifts in 19th-Century British Grammar Writing: A Network of Texts and Authors." In *Norms and Conventions in the History of English*, edited by Birte Bös and Claudia Claridge, 49–71. Current Issues in Linguistic Theory. Amsterdam: John Benjamins.
- **Busse**, Beatrix, Kirsten **Gather**, and Ingo **Kleiber**. forth. "A Corpus-Based Analysis of Grammarians' References in 19th-Century British Grammars." In *Variation in Time and Space: Observing the World through Corpora*, edited by Anna Cermakova and Markéta Malá. Diskursmuster - Discourse Patterns 20. Berlin: De Gruyter.