Utilising ANNIS for search and analysis of historical data

Stephan Druskat    Thomas Krause    Carolin Odebrecht

Institut für deutsche Sprache und Linguistik
Humboldt-Universität zu Berlin

Reuse or New Development: sustainability of resources and tools for multi-facetted historical data and languages.
FORGE 2016
Outline

1. Challenges for a query system for historical corpora
2. Development process of ANNIS (also concerning sustainability)
3. Case study: Using ANNIS for Coptic corpora
Challenges for a query system for historical corpora

- support for a wide range of scripts with Unicode
Challenges for a query system for historical corpora

- support for a wide range of scripts with Unicode
  - store data in a supported encoding (UTF-8, UTF-16, etc.)
  - search strings (including regular expression engine)
  - display in front-end
Challenges for a query system for historical corpora

- support for a wide range of scripts with Unicode
  - store data in a supported encoding (UTF-8, UTF-16, etc.)
  - search strings (including regular expression engine)
  - display in front-end
- representation of both the original script, transliterations and translations
Challenges for a query system for historical corpora

- support for a wide range of scripts with Unicode
  - store data in a supported encoding (UTF-8, UTF-16, etc.)
  - search strings (including regular expression engine)
  - display in front-end
- representation of both the original script, transliterations and translations
  - multiple (aligned) tokenizations
  - facsimiles
Challenges for a query system for historical corpora

- support for a wide range of scripts with Unicode
  - store data in a supported encoding (UTF-8, UTF-16, etc.)
  - search strings (including regular expression engine)
  - display in front-end
- representation of both the original script, transliterations and translations
  - multiple (aligned) tokenizations
  - facsimiles
- representation of different linguistic theories
Challenges for a query system for historical corpora

- support for a wide range of scripts with Unicode
  - store data in a supported encoding (UTF-8, UTF-16, etc.)
  - search strings (including regular expression engine)
  - display in front-end
- representation of both the original script, transliterations and translations
  - multiple (aligned) tokenizations
  - facsimiles
- representation of different linguistic theories
  - multiple layers of different kinds of annotations (grids, constituent trees, dependencies trees, ...)
  - no fixed annotation scheme
Are these challenges unique?

- these challenges will arise when dealing with almost any historical corpus
- examples: RIDGES Herbology [2], Coptic SCRIPTORIUM [4], Referenzkorpus Altdeutsch [3], . . .
- enough overlap in problems to justify a common solution
Are these challenges unique?

- these challenges will arise when dealing with almost any historical corpus
- examples: RIDGES Herbology [2], Coptic SCRIPTORIUM [4], Referenzkorpus Altdeutsch [3], . . .
- enough overlap in problems to justify a common solution

Does developing a corpus search system together as community solve some of the sustainability issues we have with academic software?
ANNIS [1] origins

- originally developed for the needs of the Sonderforschungsbereich 632 Informationsstruktur - project from 2003 to 2015
ANNIS [1] origins

- originally developed for the needs of the Sonderforschungsbereich 632 Informationsstruktur - project from 2003 to 2015
- intended as common corpus search system for the different corpora developed in the SFB 632
  - newspaper corpora,
  - historic corpora,
  - spoken language corpora,
  - ...
ANNIS [1] origins

- originally developed for the needs of the Sonderforschungsbereich 632 Informationsstruktur - project from 2003 to 2015
- intended as common corpus search system for the different corpora developed in the SFB 632
  - newspaper corpora,
  - historic corpora,
  - spoken language corpora,
  - ...
- goal: develop a corpus-independent multi-layer corpus search tool
 ANNIS technology

- web-based, using the Vaadin Framework ([https://vaadin.com/home](https://vaadin.com/home))
- written in the Java Programming Language
- Maven build system ([https://maven.apache.org/](https://maven.apache.org/))
- relational database PostgreSQL ([https://www.postgresql.org/](https://www.postgresql.org/)) is used for the actual search
- split into web front-end and REST service
- new visualizations and exporters can be added as plug-ins
Sustainability aspects: license

- every project (software and corpora) needs a proper license
  - clarifies what is allowed and what isn’t
  - no explicit license means “can’t be used at all”
Sustainability aspects: license

- every project (software and corpora) needs a proper license
  - clarifies what is allowed and what isn’t
  - no explicit license means ”can’t be used at all”
- Open source under the permissive Apache License 2.0
- not limiting commercial use
- possible to fork under own license
- clarifies that any submitted contribution is licensed under the Apache License 2.0 per default
Sustainability aspects: license

- every project (software and corpora) needs a proper license
  - clarifies what is allowed and what isn’t
  - no explicit license means “can’t be used at all”
- Open source under the permissive Apache License 2.0
- not limiting commercial use
- possible to fork under own license
- clarifies that any submitted contribution is licensed under the Apache License 2.0 per default

Goal:
Allow people to use your software without any restriction and provide a clear legal path of how to maintain the project even without participation of the original copyright holders.
Sustainability aspects: project hosting

- source code hosted in a public GitHub project (https://github.com/korpling/ANNIS/)
  - allows pull requests and reporting issues as possibility to contribute to the project
  - 13 different developers contributed in total
  - 67 pull requests
  - 464 issues tracked
  - managing them is work and needs resources
- documentation and web-site hosted on GitHub infrastructure
- binaries published in Maven repository

Goal: Make infrastructure as independent from current maintainers as possible.
Sustainability aspects: project hosting

- source code hosted in a public GitHub project (https://github.com/korpling/ANNIS/)
- allows pull requests and reporting issues as possibility to contribute to the project
  - 13 different developers contributed in total
  - 67 pull requests
  - 464 issues tracked
  - managing them is work and needs resources
Sustainability aspects: project hosting

- source code hosted in a public GitHub project (https://github.com/korpling/ANNIS/)
- allows pull requests and reporting issues as possibility to contribute to the project
  - 13 different developers contributed in total
  - 67 pull requests
  - 464 issues tracked
  - managing them is work and needs resources
- documentation and web-site hosted on GitHub infrastructure
Sustainability aspects: project hosting

- source code hosted in a public GitHub project (https://github.com/korpling/ANNIS/)
- allows pull requests and reporting issues as possibility to contribute to the project
  - 13 different developers contributed in total
  - 67 pull requests
  - 464 issues tracked
  - managing them is work and needs resources
- documentation and web-site hosted on GitHub infrastructure
- binaries published in Maven repository
Sustainability aspects: project hosting

- source code hosted in a public GitHub project (https://github.com/korpling/ANNIS/)
- allows pull requests and reporting issues as possibility to contribute to the project
  - 13 different developers contributed in total
  - 67 pull requests
  - 464 issues tracked
  - managing them is work and needs resources
- documentation and web-site hosted on GitHub infrastructure
- binaries published in Maven repository

Goal:
Make infrastructure as independent from current maintainers as possible.
Sustainability aspects: people involved

- encourage participation from outside parties for certain features
Sustainability aspects: people involved

- encourage participation from outside parties for certain features
  - collect feature ideas
  - generalize different use-cases into common set of features
  - coordinate developers
- provide help for new developers to implement these features
- remove technical obstacles for collaboration when we become aware of them (build system, documentation etc.)
- big question: who can ensure coordination for a longer time?

Goal: Foster a community of developers and users with interest in the project and make participation as easy as possible.
Sustainability aspects: people involved

- encourage participation from outside parties for certain features
  - collect feature ideas
  - generalize different use-cases into common set of features
  - coordinate developers
- provide help for new developers to implement these features
Sustainability aspects: people involved

- encourage participation from outside parties for certain features
  - collect feature ideas
  - generalize different use-cases into common set of features
  - coordinate developers
- provide help for new developers to implement these features
- remove technical obstacles for collaboration when we become aware of them (build system, documentation etc.)
Sustainability aspects: people involved

- encourage participation from outside parties for certain features
  - collect feature ideas
  - generalize different use-cases into common set of features
  - coordinate developers
- provide help for new developers to implement these features
- remove technical obstacles for collaboration when we become aware of them (build system, documentation etc.)
- big question: who can ensure coordination for a longer time?
Sustainability aspects: people involved

- encourage participation from outside parties for certain features
  - collect feature ideas
  - generalize different use-cases into common set of features
  - coordinate developers
- provide help for new developers to implement these features
- remove technical obstacles for collaboration when we become aware of them (build system, documentation etc.)
- big question: who can ensure coordination for a longer time?

**Goal:**
Foster a community of developers and users with interest in the project and make participation as easy as possible.
Case Study: Coptic Scriptorium

- joint project of Carrie Schroeder (University of the Pacific) and Amir Zeldes (Georgetown University)
- wanted to use ANNIS as their search system
Case Study: Coptic Scriptorium

- A joint project of Carrie Schroeder (University of the Pacific) and Amir Zeldes (Georgetown University)
- Wanted to use ANNIS as their search system
- ANNIS included most required features but was also missing some:
  - Unicode for search ✓
  - Multi-layer ✓
  - Multiple tokenization ✓
  - Web-fonts for displaying coptic script ×
  - Re-creating the appearance of the facsimiles without using the actual images ×
Demo

https://corpling.uis.georgetown.edu/annis/?id=bff7712f-b60d-4d58-876e-483048e79eb5
https://corpling.uis.georgetown.edu/annis/?id=bff7712f-b60d-4d58-876e-483048e79eb5

- virtual keyboard
- Unicode search
- visualization layers (grid), including custom font
- HTML document visualization
HTML visualizer: idea

- facsimile not publicly available
- sub-set of graphical TEI annotations
- specialized visualizer needed that renders these graphical annotations
  - substantial overhead to write visualizer for a single corpus
  - different historical corpora (not just based on TEI) would need the same kind of visualizer but have different annotations

https://corpling.uis.georgetown.edu/annis/?id=c27b6810-556a-42bb-89ee-46e5046a3ded
HTML visualizer: idea

- facsimile not publicly available
- sub-set of graphical TEI annotations
- specialized visualizer needed that renders these graphical annotations
  - substantial overhead to write visualizer for a single corpus
  - different historical corpora (not just based on TEI) would need the same kind of visualizer but have different annotations

Idea
To write a generic visualizer that maps the structure of the span annotations (grid) to HTML tags with configurable rules and custom CSS.

https://corpling.uis.georgetown.edu/annis/?id=c27b6810-556a-42bb-89ee-46e5046a3ded
HTML visualizer: implementation

- joint effort of Amir Zeldes (Georgetown) and Humboldt-Universität zu Berlin
- common discussions over design, with influences from the Coptic Scriptorium project and the RIDGES corpus
- implementation split up into several smaller features
- different features implemented by different developers from the working groups
HTML visualizer: improvement

- early releases with basic features and incremental updates for new features
- new features mostly driven by new corpora or new perspectives on how to use them
  - visualizations can be embedded to other web-sites:
    - https://corpling.uis.georgetown.edu/annis/?id=e99c10c9-f814-4be0-b71a-40dda541bcca,
    - https://korpling.org/annis3/?id=eb2d3696-d69b-4d7e-82f2-396c78ca01ba
  - automatically generated links to dictionaries with a template system http://data.copticscriptorium.org/texts/ap/ap004poemen65/norm
Conclusion

- coordinating developers from different teams is possible even in academic research projects
- generalizing feature ideas helps to increase the impact, can be used by more than one corpus
- sharing tools possible when tools are generic and specific enough at the same time
- technical and legal issues must be tackled
References I

