Maximilian Eibl, Martin Gaedke. (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 1179

To GUI or not to GUI?

Nils Reiter[1], Jonas Kuhn[1], Marcus Willand[2]

Abstract: This paper focuses on the interface question: How do literary scholars interact with their texts? We discuss the pros and cons of different options and then describe the research workflow that we are employing in the projects CRETA and QuaDramA. We believe that DH projects too often take the need for graphical user interfaces for granted and want to establish that other options have their pragmatic and conceptual merit.

Keywords: Graphical User Interface, Literary Studies, Natural Language Processing, Reflected Text Analytics

1 Introduction

The decision on the concrete workflow that a DH project instantiates is typically made very early in a project. This has institutional reasons, as detailed workflows are a standard demand for funding applications. In many projects, those workflows involve one or more graphical user interfaces (GUI). These GUIs are often tailored to the specific needs and research data of the project and developed during the project run time. Our observation is that this is rarely questioned or consciously decided. In the projects QuaDramA and CRETA, we continuously evaluate various options in this regard and try to learn from past experiences.

The arguments in this article focus on the analytical use of digital methods in the humanities[3], in particular on the use of natural language processing (NLP) and literary texts. We therefore do not claim that what we present is ubiquitously applicable in the DH, although we believe some of the insights can be applied to other projects.

We will briefly introduce the two projects that we are using as exemplary cases here. Section 2 discusses the main arguments for and against graphical user interfaces. Sections 3 and 4 describe how the workflows in these projects are established, giving different examples on realizations of the arguments presented here. Section 5 concludes.

[1] Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Pfaffenwaldring 5b, 70569 Stuttgart, nils.reiter@ims.uni-stuttgart.de, jonas.kuhn@ims.uni-stuttgart.de

[2] Institut für Literaturwissenschaft, Neuere Deutsche Literatur II, Universität Stuttgart, Keplerstraße 17, 70174 Stuttgart, marcus.willand@ilw.uni-stuttgart.de

[3] In contrast to, e.g., data preparation (OCR) or infrastructure development.

doi:10.18420/in2017_119

CRETA One of the research avenues in the Center for Reflected Text Analytics (CRETA)[4] is the analysis of relations between characters in narrative texts, in particular the middle high German Arthurian novel *Parzival* and Goethe's *Sorrows of the Young Werther*. As a basis, we are extracting co-occurrence networks from both texts. Extracting such networks from dramatic texts is technically straightforward [TFK15], the meaningfulness of such networks on narrative texts depends on the notion of co-occurrence and of its technical realization.

QuaDramA The goal of QuaDramA[5] – Quantitative Drama Analytics – is the large-scale quantitative analysis of German dramatic texts from 1740-1920. We are therefore aiming for the semantic analysis of a linguistically preprocessed corpus. Most importantly, the semantic analysis needs to be conducted not only on 'plain' text level: Many of the scholarly research questions are about dramatic figures and their speech. We need to be able to distinguish the text content of various figures, but – depending on the exact question – also aggregate figure from various dramatic texts, in order to investigate classes of figures like *fathers* or *kings*.

2 GUI: A not so Binary Decision

In addition to the general, methodological and theoretic challenges of typical DH projects [KR15], an important and re-appearing question is how to concretely organize the interaction between NLP methods and literary research questions. Arguably, this question is not primarily a technical one, but also touches organizational and methodological considerations. To discuss the technical aspects without taking organizational facts and methodological assumptions into account would be a mistake and could – with good reason – be judged as 'technology-driven'.

In our cases, the concrete questions are: How do literary scholars use and access the tools that NLP researchers develop and provide? What data formats do we use to exchange data? What kind of user interface will literary scholars use to investigate their questions?

We will, for the sake of brevity and clarity, distinguish two broad options: Developing a GUI or not. Many projects (as the ones discussed below) actually do both in different stages, or employ nuanced and mixed forms in between. In addition, developing a GUI entails a wide range of sub-sequential choices (that might have impact on the workflow within a project): Web-based or desktop, centralized or distributed data storage, based on external infrastructures or not, ... On the other hand, deciding against a graphical user interface leads to a lot of follow-up questions as well: What user interface do we use instead? How do literary scholars interact with the their data? What do they have to know?

[4] https://www.creta.uni-stuttgart.de/index.php/en/

[5] https://quadrama.github.io/index.en

2.1 Developing a GUI

The most obvious and often cited advantage of implementing a graphical user interface in the context of DH is that scholars without additional training are able to use it. GUIs thus lower the entry bar for participating in DH research and potentially allow more scholars to investigate their research questions empirically (or digitally, whatever that means). Depending on the research question(s), some projects might be aiming for communicating with pupils, the interested public (citizen science), to collect data or research perspectives. In these cases, a graphical user interface might also serve as a motivational tool.

However, while this argument is so appealing, it is not without pitfalls. Having a GUI does not free a scholar from knowing about the relevant basics of the machinery. In order to provide a literary interpretation that takes quantified findings into account *and* that would stand in the eyes of a critical reader, scholars *also* need to be able to explain why an edge in a network became thick or a word in a word cloud red – any why this matters. This requires a certain understanding of the machinery, which GUIs tend to hide away. In this respect, tools with a GUI risk being run as a 'black box', with little or no understanding of the underlying assumptions and internal workings. In extreme cases, this makes it very easy to fiddle with the parameters of a tool until the resulting plot fits with the expectation.

It is not a new insight that GUIs sometimes also lead to misinterpretations. In the context of DH, however, the danger becomes even greater, as the users of GUIs often have little experience in reading quantitative data, have usually no way of verifying independently what they see and are used to presume meaning on the subject of their interest: "The worst dangers [of data mining] may lie in the humanist's ability to interpret nearly any result, projecting his or her own biases into the outcome of an experiment[. . .]" [SP08, p. 409]. One might argue that GUIs should be designed in such a way that they do not invite wrong interpretations, that they are not misleading. This is certainly true, but the reality is that most DH projects do not have access to knowledge about creating GUIs, let alone any experience.

As research generally is a creative process, some of the most interesting research questions only develop over time. Relying on a GUI for every experiment or experimental setup increases the time lost between having an idea for an experiment and actually conducting the experiment. Parameterization of GUIs can mitigate this to a certain extent, but this only goes so far, as interesting questions and ideas are difficult if not impossible to anticipate.

2.2 Direct Data Access for Scholars

Obviously, the alternative to GUIs is not to have no interface at all. Some interface between scholars and their subject of interest is needed. What we advocate here is to expose scholars to the 'messiness' of working with the actual data, whatever form it might have. At first, this heightens the entry bar into DH research. Scholars need to learn about data formats, conversion between them, handling of large files or file collections, use of a statistical

analysis software, ... All this takes time and some investment, in particular at the beginning of a project. It also might entail the need for installing more software on more computers, which creates administrative overhead.

On the other side, scholars who are able to work with data directly (with tools that are already existing) are able to conduct experiments by themselves (within certain limits). This not only decreases the development time needed between idea and experiment, it also make scholars more independent from computer scientists or NLP researchers, and lets the latter focus on the real technical, methodological or engineering challenges.

Learning a scripting language or general purpose analysis environment (such as RStudio, the python shell or CQP) also leads to valuable skills that benefit the involved scholars in future projects. In this way, investing the time to learn basics of scripting languages is more sustainable than getting used to a task-specific user interface.

3 CRETA Workflow

In order to extract character relation networks from narrative texts, CRETA employs a three-step workflow: Annotation of entity mentions in texts, segmentation of the text and extraction of and interaction with networks. The annotation of entity mentions closely resembles the NLP task of named entity recognition and can be addressed with an NLP workflow (see, for instance [NS07] for survey). Segmentation, however, is different: Not only is the task structurally more complex, it is also closer related to the specific research question literary scholars have, and needs to take them into account.

Therefore, we employ tools to allow literary scholars to experiment with different kinds of segmentation: Based on linguistic concepts (e.g., sentences), structural information (e.g., a letter in *Werther* or a 30-verse block in *Parzival*) or content-related information (an 'episode', as given by the plot of the narrative). Content based segmentation is based on textual annotation, structural information is given in the text (depending on file format), and linguistic segmentation is created based on linguistic preprocessing.

Technically, this is achieved by a combination of different tools. For entity mention detection, we annotated a corpus and trained sequence labelling models using ClearTk [BOB14] that can detect entity mentions on new texts. The segmentation interface allows specification of regular expressions (as segment boundaries) or pre-existing base annotations (automatically detected sentences or manually annotated plot units) and export of the resulting co-occurrence networks into GraphML. This further allows inspection of and interaction with the networks in a number of tools, for instance Gephi[6].

It is important to note that only some of these steps involve graphical user interfaces (e.g., the annotation tool). In order to generate an appropriate segmentation, however, scholars

[6] https://gephi.org

working on the texts also work on the regular expressions to actually segment the texts. If an expression (e.g., to capture date expressions that mark letters in *Werther*) over-generates, they debug it themselves.

4 QuaDramA Workflow

The technical workflow in QuaDramA is split into two major parts, and our scholarly partners are only involved in the second.

NLP Components NLP components are – as usual – arranged in a processing pipeline (higher level processing steps depend on lower level steps). We employ the pipeline framework Apache UIMA for this purpose and develop drama-specific components as UIMA components. Existing NLP components (as in DKpro [EdCG14]) are reused as much as possible. New components are to a large extent based on the machine learning framework ClearTk [BOB14]. As the project continues, NLP components are adapted and improved. From time to time, we re-process the entire corpus, and distribute the resulting annotated dramatic texts to all project partners. In this setup, the NLP pipeline is used to prepare data for the analysis.

Data Analysis Data analysis in QuaDramA is done with the R IDE RStudio. This allows quick analysis of large data sets, built-in plotting facilities and re-use of existing text/data analysis components available from within the DH community (e.g., stylo [ERK16] or syuzhet[7]). Analysis steps that are done repeatedly (e.g., analyzing the topical distribution of the figure speech) are wrapped into functions, which are combined in the drama-specific R package `DramaAnalysis`[8]. The use of R for data analysis by the literary scholars in the project does not work out of the box, however. It is clear that this only works if the necessary time is invested to learn some of the basics of R (and, in turn, some basic programming concepts). This, however, is supported by an increasing amount of handbook literature specifically for the DH community (e.g., [AT15]).

By using the console of a full-fledged programming language, we are also able to 'scale' with the requirements throughout the project runtime. Instead of re-developing some user interface, scholars can gradually use more complex analyses, without switching the technical or conceptual environment.

5 Conclusions

The appropriate visualization of data properties is also under discussion in the information visualization/visual analytics community [Ke08, KTM09], and our position is not in

[7] `http://www.matthewjockers.net/2015/02/02/syuzhet/`
[8] `https://github.com/quadrama/DramaAnalysis`

opposition. We are not arguing against visualization – in fact, powerful visualizations are a reason for opting for R. But we are arguing against 'blind visualization', without realizing what is visualized and how.

Modeling the workflow in a concrete DH project should not be taken lightly, and poses a number of challenges. Some are related to the specific nature of digital humanities, while others are more general management challenges. We have argued that not providing a unified graphical user interface is an option that – in addition to reducing development effort – is beneficial to the project goals, because technical details, that often have a profound influence on the analysis results, are not hidden away. In addition, it is beneficial to the development of the digital humanities as a field, as involved scholars gain experience with tools that can be re-used in other projects.

References

[AT15] Arnold, Taylor; Tilton, Lauren: Humanities Data in R. Springer International Publishing, 2015.

[BOB14] Bethard, Steven; Ogren, Philip; Becker, Lee: ClearTK 2.0: Design Patterns for Machine Learning in UIMA. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). European Language Resources Association (ELRA), Reykjavik, Iceland, pp. 3289–3293, 5 2014.

[EdCG14] Eckart de Castilho, Richard; Gurevych, Iryna: A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In: Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, pp. 1–11, August 2014.

[ERK16] Eder, Maciej; Rybicki, Jan; Kestemont, Mike: Stylometry with R: a package for computational text analysis. R Journal, 8(1):107–121, 2016.

[Ke08] Kerren, Andreas; Stasko, John T.; Jean-Daniel; North, Fekete Chris, eds. Information Visualization, volume 4950 of Lecture Notes in Computer Science. Springer, 2008.

[KR15] Kuhn, Jonas; Reiter, Nils: A Plea for a Method-Driven Agenda in the Digital Humanities. In: Proceedings of Digital Humanities 2015. Sydney, Australia, June 2015.

[KTM09] Kielman, Joe; Thomas, Jim; May, Richard: Foundations and Frontiers in Visual Analytics. Information Visualization, 8:239–246, 2009.

[NS07] Nadeau, David; Sekine, Satoshi: A survey of named entity recognition and classification. Lingvisticæ Investigationes, 30(1):3–26, 2007.

[SP08] Sculley, D.; Pasanek, Bradley M.: Meaning and mining: the impact of implicit assumptions in data mining for the humanities. Literary and Linguistic Computing, 23(4):409–424, 2008.

[TFK15] Trilcke, Peer; Fischer, Frank; Kampkaspar, Dario: Digital Network Analysis of Dramatic Texts. In: DH2015 Conference Abstracts. Sydney, Australia, 2015.