Originally published in: International Journal of Digital Curation vol. 15 (2020) no. 1, 13 pp.
DOI: *https://doi.org/10.2218/ijdc.v15i1.601*

IJDC | *Research Paper*

Towards Continuous Quality Control for Spoken Language Corpora

Hanna Hedeland
University of Hamburg

Anne Ferger
University of Hamburg

Abstract

This paper describes the development of a systematic approach to the creation, management and curation of linguistic resources, particularly spoken language corpora. It also presents first steps towards a framework for continuous quality control to be used within external research projects by non-technical users, and discuss various domain and discipline specific problems and individual solutions. The creation of spoken language corpora is not only a time-consuming and costly process, but the created resources often represent intangible cultural heritage, containing recordings of, for example, extinct languages or historical events. Since high quality resources are needed to enable re-use in as many future contexts as possible, researchers need to be provided with the necessary means for quality control. We believe that this includes methods and tools adapted to Humanities researchers as non-technical users, and that these methods and tools need to be developed to support existing tasks and goals of research projects.

Received 07 December 2018 ~ *Revision received* 14 October 2019 ~ *Accepted* 20 January 2020

Correspondence should be addressed to Hanna Hedeland. Email: hanna@hedeland.org

An earlier version of this paper was presented at the 13th International Digital Curation Conference.

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by the University of Edinburgh on behalf of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

Introduction

This paper presents the development of a systematic approach to research data management and data curation for linguistic resources, in particular spoken language corpora, with the specific aim of enhancing quality control and quality assurance. Currently, research data created within various settings require an often lengthy curation process before it can be disseminated for re-use within similar and different contexts. Apart from data completeness and consistency, an important aspect of this data curation is to ensure metadata and further descriptions make the data understandable to re-users. The thorough curation process carried out at the Hamburg Centre for Language Corpora (HZSK)[1], a research data centre specializing in language corpora with a thematic focus on linguistic diversity, is based on a software system for quality control, which is one aspect of the quality assurance work described within this paper.
Apart from the internal processes and workflows within the centre, the paper thus also focuses on the cooperation with external research projects from various disciplines and the technological, methodological and administrative challenges that come with research data management in such a setting. The support during the project phase has proved highly important for the outcome of the project and the quality of the research data, i.e. the language corpora deposited with the centre. The paper also reports on the experiences gained with the approach presented in the paper within the long-term project INEL ("Grammatical Descriptions, Corpora and Language Technology for Indigenous Northern Eurasian Languages") (Arkhipov and Däbritz, 2018). While the resource type considered in this paper is highly specific, the general approach and the challenges of cooperative settings are applicable for most contexts in which research data is created or enriched manually for analysis and questions of quality management have still to be answered from the various participants' perspectives. The approach exploits concepts and strategies from software development, such as distributed versioning, issue tracking and extensive testing, for the creation of high quality digital resources. By working towards continuous quality control and continuous integration, we aim to prevent the high curation costs often involved in making spoken language corpora from research projects re-usable in a wider context. Our experience showed a significant decrease of working hours needed to prepare and curate the data before making it available to an audience. In a comparison of the different stages of adapting our framework, the workload of the publication preparation process could be significantly reduced. During the first raw implementation phase the amount of time could be decreased by 30% in comparison to the work before.

Related Work

Following the general societal interest in programming and agile methods, and the recent discussions on reproducibility of research (cf. Baker, 2016), the idea to adapt software development concepts such as distributed version control, e.g. using Git[2], and continuous integration (Fowler, 2006) for research data management has been implemented to various extents for several settings. This development is also recognized

1 Hamburg Centre for Language Corpora: https://corpora.uni-hamburg.de/hzsk/
2 Git: https://git-scm.com/

by providers of technical platforms such as GitLab[3] and is spreading into the research communities (cf. Vuorre and Curley, 2018). Some work relevant to our approach are the Cross-Linguistic Linked Data project (Forkel, 2014), in which GitHub[4] and Zenodo[5] are used to manage and publish cross-linguistic databases, the continuous integration approach for digital editions from the Perseus Digital Library (Almas and Clérice, 2017) and the agile corpus creation approaches described by Voorman and Gut (2008) and Druskat et al. (2016). However, when it comes to spoken language corpora, existing Git hosting solutions such as GitHub are, in most cases, not an option due to privacy and data protection issues.

Efficient versioning of the media files is yet another resource type specific problem, due to performance issues arising when adding binary files to a versioning framework created for code or simple text files. While several solutions exist for Git a custom solution using git, sparse-checkout needed to be implemented for the special resource type which will be explained in a following section (see *Specific Git Solutions*).

However, the main difference between most related work and the setting described in this paper pertains to the participants and the roles in the curation process, in which the research data is prepared for publication. While internal creation of resources or curation of existing complete legacy data sets allow for a controlled and systematic process, it was found that the deposit of corpora from external research projects created manually directly with the dedicated EXMARaLDA[6] software suite (Schmidt and Wörner, 2014), which is developed at the centre, posed other challenges. Most of the projects apply qualitative methods for which data consistency and structural correctness are of less importance, and project conventions and schemas are usually refined repeatedly according to research findings. Problems with data consistency and structural errors not only limit the options for re-use, making querying of the data unreliable, but also pose a problem for automatic conversion into future data formats according to the centre's preservation plan. Although EXMARaLDA features basic checks for problems with syntax, structure and metadata consistency, the required flexibility of the tools still results in inconsistent data. Those inconsistencies can consist of spelling errors in the data itself as well in the metadata, or differences in the internal structure of the files. The greatest challenge thus turned out to be how to enable external projects to actively anticipate re-use, i.e. to create data that requires fewer working hours of curation before its publication, by allowing for continuous quality control throughout the project duration. This question is addressed for research data management in general by the CONQUAIRE project[7] (Ayer et al., 2017), which are developing a framework based on GitLab with specific unit tests for various resource types, though not for the specific requirements of the resource described in this paper.

Resource Type Specific Characteristics

Despite recent advances in natural language processing, artificial intelligence and related fields, the creation of linguistic research data still relies to a large extent on manual processes. This also holds true for spoken language corpora, since transcription of spoken language cannot be carried out automatically for under-described and non-

3 GitLab: https://about.gitlab.com/2017/08/25/gitlab-and-reproducibility/
4 GitHub: https://github.com/
5 Zenodo: https://zenodo.org/
6 EXMARaLDA: http://exmaralda.org/de/
7 CONQUAIRE: https://conquaire.uni-bielefeld.de

standard linguistic varieties, or any conversation containing heavily overlapping speech. Human interpretation of the signal and the context is also required for many kinds of annotation, i.e. further information or categories from a fixed set added to the transcribed text. Furthermore, the collection and organization of metadata cannot be automatized, making the creation of such data sets very costly and time-consuming.

At a closer look (cf. Figure 1), the complexity of the resource type described in this paper becomes clear: spoken language corpora comprise audio and/or video recordings of spoken language, corresponding time-aligned textual representations (transcripts), often enriched with further analytical information (annotation), and in many cases there is also additional material relevant to the recording situation, to a speaker, or to the corpus as a whole. As an example, the spoken language corpora created within the INEL project often additionally include scans of (hand-)written documents. Furthermore, the annotation, i.e. the analytic information in the transcripts, was added manually and includes morpheme segmentation of words, various types of grammatical information, syntactic functions, notes on language use (code switching), and other linguistically relevant information. The transcript files also contain translations into Russian, English and German.

At the Hamburg Centre for Language Corpora, all spoken language corpora (and some written) are represented using EXMARaLDA, a system of open source software tools and XML data formats. Whereas the transcript files are created with the transcription and annotation editor (Partitur-Editor), the metadata for each of the abstract and physical items of the corpus, e.g. on a setting or a speaker, but also a transcript or a recording, is created with the EXMARaLDA corpus and metadata manager (Coma) and represented by its own XML format. In addition to the metadata collected within the original project setting, a resource published via the HZSK Repository[8] also needs more generic and standardized metadata to ensure discoverability and comprehensibility beyond the original context. This metadata can be semi-automatically generated from the project's corpus metadata, but often human interpretation is necessary for completion. As a CLARIN[9] centre, part of the European Research Infrastructure for Language Resources and Technology, the Hamburg Centre for Language Corpora provides public standardized CMDI[10] metadata via OAI-PMH[11], but also metadata in the more widely used DC[12] and OLAC[13] formats.

8 HZSK Repository: https://corpora.uni-hamburg.de/repository
9 CLARIN: https://www.clarin.eu/
10 CMDI: https://www.clarin.eu/content/component-metadata
11 OAI-PMH: https://www.openarchives.org/pmh/
12 DC: http://dublincore.org/
13 OLAC: http://www.language-archives.org/

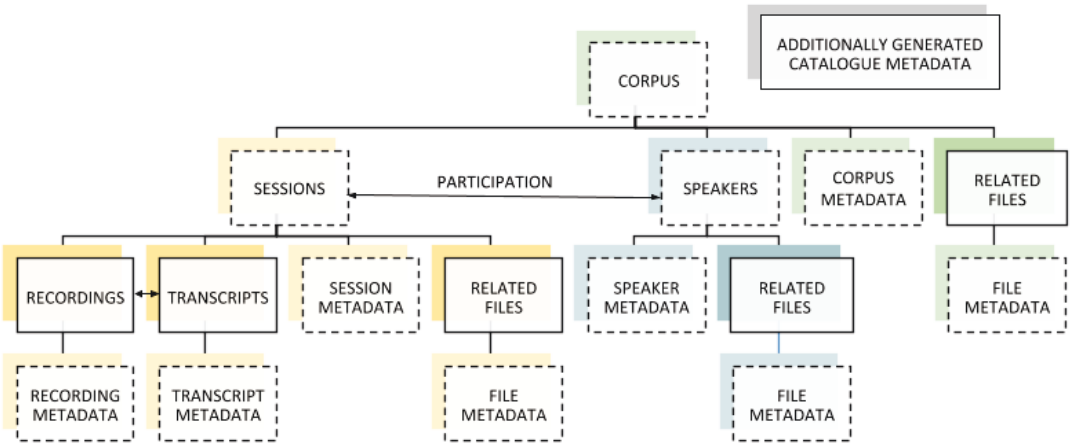

Figure 1. A spoken language corpus.

While discoverability as a requirement can possibly be tackled in a more general way, when reviewing the FAIR[14] principles for this resource type, re-usability, and in particular the "domain-relevant community standards" (R1.3.), entail high requirements on the data: a crucial characteristic of digital corpora is the possibility of querying all data and metadata in a systematic manner, which requires a structured and consistent data set. To allow for widespread re-use, the transcripts and all individual parts and conventions must be correct, consistent and well-documented. Otherwise it is impossible for an automatic process (e.g. complex linguistic queries on the data or creation of statistics) to disambiguate, for example, which words comprise the original text and which a translation, which is the set of labels attached to specific grammatical categories or which stretches of text belong to which defined speaker. At the same time, it is not possible to further standardize spoken language corpora across various research questions and theoretical frameworks – many of which follow an explorative approach and refine conventions for transcription and annotation throughout the project duration – than in terms of such generic principles as correctness, consistency and thorough documentation. To arrive at a "certification, from a recognized body, of the resource meeting community standards," as suggested in the corresponding FAIR metric[15], does not yet seem feasible for the entire heterogeneous group of researchers possibly interested in a certain data set, e.g. (various kinds of) linguists, historians, anthropologists etc., and has not yet been reliably implemented even for more specific research communities. However, since spoken language corpora might represent intangible cultural heritage, containing, for example, the last speakers of an extinct language, proper curation is a question of creating a valuable resource usable in many contexts.

Quality Management for Spoken Language Corpora

While completely automated curation based on exact specifications for all data types would be the first choice to arrive at high quality spoken language corpora, it would require unacceptable constraints on the creation of the data, since it would require that

14 FAIR Principles: https://www.force11.org/group/fairgroup/fairprinciples
15 FAIR Metrics R1.3: https://purl.org/fair-metrics/FM_R1.3

projects specify the complete structure and conventions for the transcripts and metadata before starting to work with the data. This is not only impossible for projects documenting lesser researched languages, but also generally not feasible for most research projects within the Humanities employing qualitative methods, where convention and design decisions can partly only be made after data has been collected and reviewed. Thus, even the existence of a detailed research data management plan does not imply that project specific decisions and conventions are used as planned and not changed because of insights generated by the reviewed data. Therefore, quality control and documentation need to be performed continuously throughout the project duration, and further developed into reliable quality assurance methods. This would make it possible to avoid extensive data curation carried out by staff in research data centres, who are on the one hand specialists in data curation, but have no knowledge of the individual projects and only access to explicit information provided by the researchers. Enabling the data creators to do most of the curation themselves would mitigate the high cost and inevitable information loss of subsequent data curation. Another common source of various problems within data management in research project is parallel work of different collaborators on the same data set. This problem has been solved in the field of software development. The approach therefore seeks to adapt the concepts used within software development for the described setting. To enable curation performed by the non-technical data creators during the data creation phase, different strategies using distributed version control, formalized workflows and automated checking and fixing mechanisms for corpus data were implemented.

Preliminary Work

Before the implementation of the framework, some preliminary work was necessary in order to gain more control over the data and the workflows relevant to its creation and curation.

Distributed version control with Git

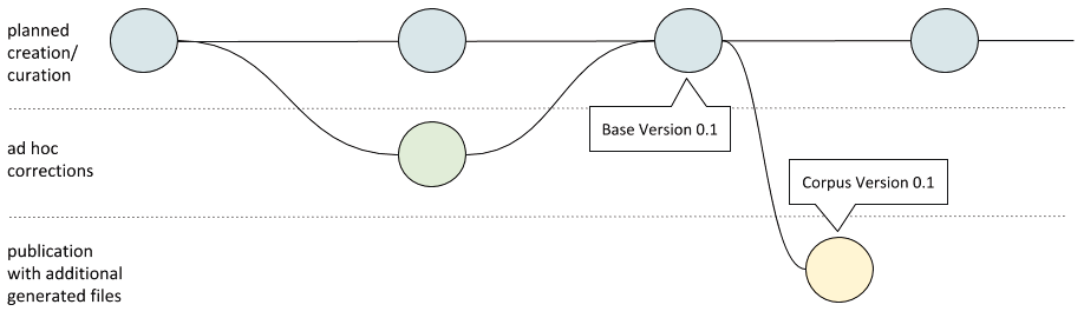

Figure 2. Git workflow.

In a first step, Git was introduced for distributed version control and to model project workflows using basic branching principles as displayed in Figure 2. Apart from the advantages of basic version control, Git facilitates collaborative workflows, as files from different collaborators can be compared and merged. Furthermore, changes can be systematically undone, e.g. inconsistencies introduced by a specific collaborator. The definition and naming of versions in the active curation workflow helps overcoming

diverse issues, like identifying states of work on which specific research was conducted, or differentiating between raw, curated and published data sets. The branching workflow was also adapted for the spoken language corpus data curation by using different branches for the planned development and curation by various collaborators working simultaneously on the corpus data. This was done by using a dedicated branch for publishing, including automatically generated publication formats, and, where necessary, additional branches for ad hoc curation steps or corrections of the data. This is in accordance with project work, which is often not carried out sequentially step by step, as depicted in the data-oriented research data lifecycle, but rather simultaneously with various parts of the data spread across the active stages of the lifecycle, often depending on external circumstances such as availability of specialized staff or native speakers, or the requirement for researchers to continuously present and publish results, for which parts of the data are analysed.

While version control of spoken language corpora with Git yields benefits, it also poses difficulties for the non-technical users, i.e. mainly linguists, working with the data. Git allows for collaborative project workflows only if it is and can be used by every collaborator. To account for this, non-technical collaborators use a simple script allowing for only basic Git tasks within a single branch. However, this approach requires maintenance and support by technical staff members and leaves room for enhancement.

Formalizing the data curation workflow

While the curation itself needs to be done manually in many cases, the workflow and the consistency checks to be performed can be formalized and embedded into digital project management tools integrated with Git. To achieve this, each curation step was recorded using the issue tracking system Redmine[16] and issues gathered from various curation projects. After collecting these issues, curation workflows were designed and structured in an inductive manner and implemented as automated custom workflows in Redmine.

In our case, we use specific tracker types for the different data curation phases, which provide detailed sub-tasks for each individual action to be performed. With this automated support of the workflow, also less experienced staff members were able to perform more of the curation tasks in a more reliable manner, since the information on how to proceed had been made explicit with the list of sub-tasks as a checklist for each curation phase. Figure 3 shows examples of two different curation phases, corpus initialisation and corpus integration, comprising several sub-tasks. By modelling the workflow as Redmine issues connected to individual revisions of the data set (cf. Figure 5), the curation process was also made more transparent and accessible for depositors and future users of the data.

16 Redmine: https://www.redmine.org/

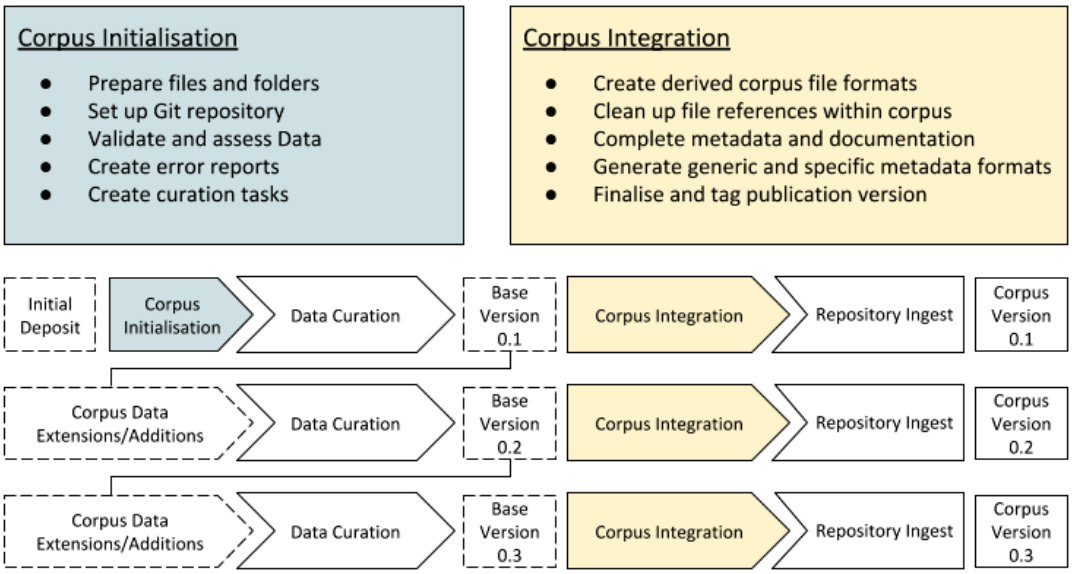

Figure 3. Custom workflows for corpus initialization and integration.

A Framework for Quality Control

The framework has several components to ensure effective quality management. Version control and formalized workflows are combined with a software solution (Corpus Services[17]) for quality control, which enables semi-automatic diagnosis through enhancements of the data and checking functionality. The workflows and practices differ corresponding to the roles of the contributors of a project. The following paragraphs will explain solutions implemented and used by the technical staff (coloured blue in Figure 4) that can be adapted to various projects. The yellow blocks show how the implemented have been adapted and further developed for use by non-technical users with support of a technical team.

Specific Git solutions

To use the framework in an efficient manner, some difficulties with Git and the linguistic data and non-technical Git users needed to be solved. Especially merging with Git, handling binary files with Git and working together with people that could not use Git called for solutions.

Lossless merging and minimizing merge conflicts in Git is not always simple, especially when working with XML data (of which the linguistic corpora described here mostly consist). While people working with XML files and Git mostly do not use Git's native automatic merge functionality (which is text- and line-based) here it can – because of the special way the XML files are created and edited. They are created via Desktop-based GUI (Graphical User Interface) tools (in this case with the EXMARaLDA software suite, compared to XML editors), which means that the XML structure can only be edited to some extent, making Git's native merge functionality usable here. Additionally, the XML files were formatted in a consistent way in the working folders of the researchers to minimize the number of merge conflicts that call for manual intervention by the technical staff.

17 Available at: http://hdl.handle.net/11022/0000-0007-D8A6-A

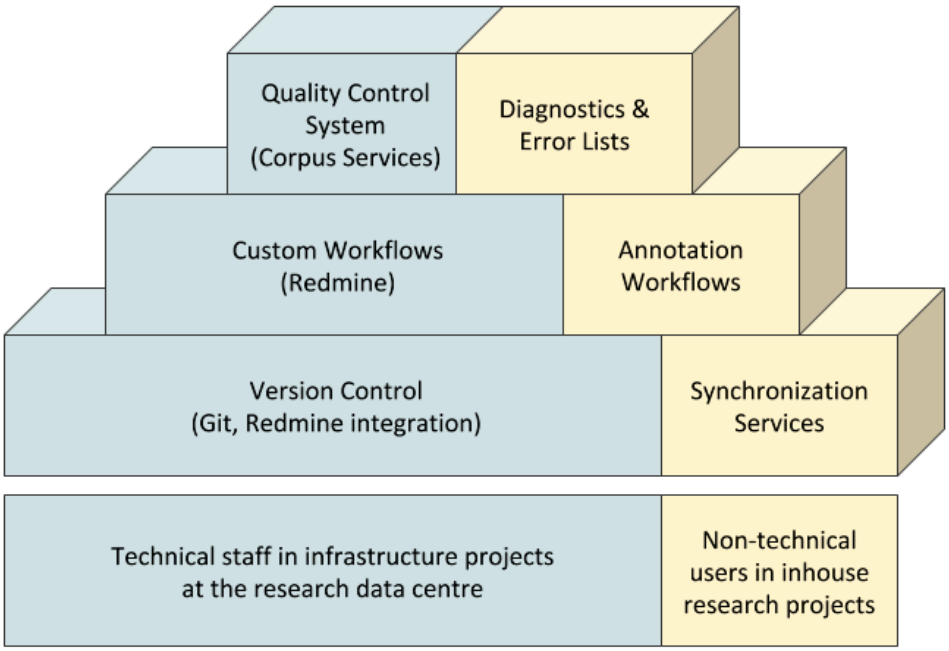

Figure 4. Components for quality management.

Versioning binary files poses another problem to many Git users. In the case of spoken language corpora, audio and video files need to be handled. When added to a corpus, those media files mostly will not be changed any further, but need to be backed up[18], have their designated location in the folder structure of the corpus (because they are linked via paths in the transcription files) and need to be supplied along with the rest of the corpus. This is done during the working phase by cloning the Git repository, so they can be used in the software tools. While there exist some solutions to handle binary files in Git (e.g. git-lfs and git-annex), in our specific case the media files need to exist at a certain file path and we could not add further complexity to our Git routines. Thus we add the binary files once but only check them out when they are really needed using git sparse-checkout to avoid performance issues.

A crucial challenge for the framework is the adaptability to researchers that can for various reasons not work with Git. To solve this, a "silent" Git workflow was created: The Git repository the researchers needed to work with was cloned onto a server accessible for both the technical team and the researcher, and a script that automatically synchronizes their changes with the bare Git repository was created. With that workflow, the researcher does not need to use Git at all but use a very simplified script. This method has some disadvantages, as no advanced Git functionality could be used by the linguists, but it also minimizes the possibilities of data loss because of the usage of wrong Git functions. When the script encounters Git conflicts it aborts the merge to make sure the researcher is not left alone with incomprehensible Git conflict files and notifies the technical staff to fix the merge conflict manually. Despite these disadvantages, the workflow highly simplifies the collaboration with researchers not able to use Git at all.

18 Backing up media files correctly also includes to monitor them to detect any unwanted alterations (corruptions of the files) and in this case return to the previous version.

Custom workflows for the approach

The framework (cf. Figure 4) contains checks with error lists in different formats and automatic fixes in the top layer. Everything that can be fixed (or be made more consistent) automatically (like using the same characters for quotation marks everywhere which simplifies further processing and conversions) will be done using the code basis of the various automatic functions. Results of checks that need fixing but cannot be fixed automatically (like mismatches of filename and its name in the metadata file or faulty links to the media file) need to be fixed by the linguists, because this often requires detailed knowledge of the data. Those cases are written into XML error lists that can be opened with the EXMARaLDA software (with automatic opening of the file at the specified location) and an HTML overview of errors. Issues that referenced the error lists were created using Redmine Custom Workflows.

Consistent implementation of checks and fixes

To enable continuous quality control and to automate parts of the curation process, all existing consistency checks and similar functionality were gathered and a dedicated software system (HZSK Corpus Services) for all kinds of corpus processing functionality relevant to curation and publication was generated. Further checks, analysis etc. can be added to the system in an efficient manner whenever problems not yet covered are discovered in the curation of new resources. The system already comprises various checks and diagnostic functionality, such as overviews of existing problems or user-friendly error lists to be used with the EXMARaLDA software, and analyses of corpus data and metadata allowing for manual error detection. It is also possible to use the system to fix problems in the resources that can be fixed automatically, but no attempts are made to fix all problems automatically.

Conclusions from Working with the Framework

In the INEL project, the new framework was used in practice within a research project for the first time, which led to different changes, enhancements and future plans.

Experiences with Git

It became clear that the levels of Git proficiency are very different among the non-technical users, i.e. the researchers, and this needs to be accounted for in the workflow. The different Git workflows for each level of proficiency (e.g. none, beginner, medium, expert) need to be more fine-grained. Additionally, the proficiency levels need to be reflected in the Git software that is used and only the required Git functions should be available there. For more advanced Git users, an option for solving Git merge conflicts in the software tools (by showing the differences in the GUI visualisation and not in the raw XML) needs to be implemented in the future.

Workflows in practice

The technical staff and the linguists worked simultaneously on the curation. Figure 5 shows an example of multiple researchers collaborating on one language corpus. The visualisation of the branches is automatically created via Redmine. While mostly adhering to the proposed workflow, it became clear that working together on identifying cases where automatic fixes can be used yielded benefits.

Figure 5. Git repository and revision overview Redmine.

Experiences with HZSK Corpus Services

Many of the automatic fixes that were necessary only became apparent during the curation. The modular structure of the code structure did allow for fast implementation of new automatic fixes, but the workflow needed to be adapted: for every automatic fix there also needed to exist a check that runs regularly on the data so the error could not be reintroduced again. Because of the collection of all the checks and fixes in one code base, all newly implemented functions could be used for future projects as well.

Furthermore, the importance of visualizing became more apparent during the application of the framework. Especially sorting and filtering in Excel-like columns (the metadata as well as the generated error lists) helped to figure out inconsistencies and possibilities for automatic fixing.

The functionality of opening error lists in the EXMARaLDA tools made manual fixing faster and easier.

Outlook

With continuous quality control implemented as automatic consistency checks and other diagnostic tests, one could argue that research data management becomes an even more difficult task, since it implies additional and more comprehensive quality requirements on the data and additional tasks to be performed by the researchers. On the other hand, a quality control framework also provides comprehensive support and thus allows for a more systematic approach to quality assurance and quality management in general for the research projects creating spoken language corpora. The experiences with the INEL

project and the extensions created for mainly non-technical users are an important step in this direction.

The aim is to introduce the framework for spoken language corpora created within external research projects, where technical support is not available to the same extent as in the INEL project, to achieve correct and consistent research data throughout the project duration. However, to make this possible, the framework has to be even more user-friendly and robust, and the use of advanced versioning systems, such as Git, needs to be simplified and adapted for researchers as non-technical users. Technical support will also still be required for more advanced tasks, which can undoubtedly pose a problem for the efficiency of the resource creation and also for the acceptance of advanced technical solutions such as Git in the Humanities.

Although not all research projects working with audio-visual data and transcripts exploit the possibilities of digital resources, but rather use the data set as a source for examples of a certain phenomenon, the responsibility of the individual research project to create resources that can be re-used in various contexts is considerable, especially in the case of language documentation or other domains related to cultural heritage. If projects fail to produce a high quality resource, future researchers are prevented from re-using data that cannot be collected in future research. It is crucial that researchers are not left alone with this responsibility, but are provided with support through technical tools and staff applying reliable methods for quality assurance and quality control tailored to the requirements of specific resource types, and thus enabling the creation of high quality digital resources.

Acknowledgements

This publication has been produced in the context of the projects CLARIN-D, funded by the German Ministry for Education and Research (BMBF) under grant number 01UG1620G, and INEL, within the joint research funding of the German Federal Government and Federal States in the Academies' Programme, with funding from the Federal Ministry of Education and Research and the Free and Hanseatic City of Hamburg. The Academies' Programme is coordinated by the Union of the German Academies of Sciences and Humanities. The HZSK Corpus Services have been developed in the context of CLARIN-D, INEL and the Hamburg Centre for Language Corpora (HZSK) at the Universität Hamburg.

References

Arkhipov, A. & Däbritz, C.L. (2018). Hamburg corpora for indigenous Northern Eurasian languages. *Tomsk Journal of Linguistics and Anthropology, 3,* 9–18. doi:10.23951/2307-6119-2018-3-9-18

Almas, B. & Clérice, T. (2017). Continuous integration and unit testing of digital editions. *Digital Humanities Quarterly, 11(4).* Available online: http://www.digitalhumanities.org/dhq/vol/11/4/000350/000350.html

Ayer, V., Pietsch, C., Vompras, J., Schirrwagen, J., Wiljes, C., Jahn, N., & Cimiano, P. (2017). Conquaire: Towards an architecture supporting continuous quality control to ensure reproducibility of research. *D-Lib Magazine, 23(1/2).* doi:10.1045/january2017-ayer

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. Survey sheds light on the 'crisis' rocking research. *Nature, 533,* 452-454. doi:10.1038/533452a

Druskat, S., Krause, T., & Odebrecht, C. (2016). *Agile creation of multi-layer corpora with corpus-tools.org.* Retrieved from Zenodo. doi:10.5281/zenodo.157166

Forkel, R. (2014). The cross-linguistic linked data project. In C. Chiarcos, J. P. McCrae, P. Osenova & C. Vertan (eds.),*3rd Workshop on Linked Data in Linguistics: Multilingual knowledge resources and natural language processing*(pp. 60–66). Available online: http://clld.org/docs/ldl2014/main.pdf

Fowler, M. (2006). *Continuous integration.* Retrieved from https://martinfowler.com/articles/continuousIntegration.html

Schmidt , T. & Wörner, K. (2014). EXMARaLDA. *Handbook on Corpus Phonology,* 402-419. doi:10.1177/2515245918754826

Voormann, H. & Gut, U. (2008). Agile corpus creation. *Corpus Linguistics and Linguistic Theory, 4(2),* 235-251. doi:10.1515/CLLT.2008.010

Vuorre, M. & Curley, J.P. (2018). Curating research assets: A tutorial on the Git version control system. *Advances in Methods and Practices in Psychological Science, 1(2),* 219-236. doi:10.1177/2515245918754826