A software-driven workflow for the reuse of language documentation data in typological studies

Stephan Druskat and Kilu von Prince
Dept. of German Studies and Linguistics
Humboldt-Universität zu Berlin
Berlin, Germany
{stephan.druskat,kilu.von.prince}@hu-berlin.de

Abstract

Existing language documentation datasets may be reused in typological research projects, if they can be evaluated for suitability. As these datasets may implement the FAIR principles insufficiently, and occur in diverse data formats, data exploration represents an alternative means of evaluation, as well as the core feature of iterative annotation-analysis cycles during the project. This paper presents a semi-automated workflow driven by a set of corpus software, which enables data exploration as part of the research process, and alleviates its cost. The presented software includes a conversion tool to deal with different formats as well as a search and analysis platform for evaluation and exploration. The authors have successfully extended the software, and implemented the presented workflow in a typological research project on the TAM systems of Melanesian languages.

1 Introduction

Corpus-based typological studies on endangered languages rely on corpus data, whose usual sources include language archives, fieldwork, and data exchange between individual researchers. Irrespective of the source, defining and compiling suitable datasets for a typological study – and working with them to answer a research question – is challenging in two respects, as available datasets
1. must be evaluated for reusability in the first place, unless they are produced during fieldwork which targets the research question;
2. may come in different data formats.

Evaluation of the reusability of a language documentation dataset for typological research includes fine-grained assessments, e.g., probing the occurrence of linguistic phenomena in the data. If the data is retrieved from language archives this is ideally supported by detailed metadata attached to the dataset, as well as suitable metadata search functionality provided by the archive web service. If the data is acquired directly from a colleague, the relevant information may also be retrieved from metadata, or through personal communication. For datasets produced during dedicated fieldwork, reusability information for a research question is inherently available. Due to differing metadata models and formats, incomplete metadata, lack of relevant metadata, and lack of suitable search functionality for archives, reusability information may not be retrievable from metadata. In short, datasets may fail to meet the FAIR Data Principles (Wilkinson et al., 2016). In this case, the assessment must be based on in-depth data exploration. Such an exploration across several datasets can be tedious and time-consuming work, especially if the datasets come in different data formats, and hence may require different tools for the exploration process.

The second challenge - having datasets in different formats - can also complicate the actual research workflow, as different tools with different analysis capabilities may be needed for different datasets. Additionally, the analyses provided by different tools may be hard to compare, or even incompatible.

We propose that the challenges presented above can be mastered in a workflow based on a set of corpus linguistics software, corpus-tools.org (Druskat et al., 2016). The software set can be used to convert both the dataset candidates that should be evaluated (‘dataset candidates’) and the actual datasets to be analysed during a study (‘research datasets’) to a single format, in which they can be imported in the included search and analy-

1With respect to metadata, datasets may fail to meet the reusability principle by not providing “(meta)data [that are] richly described with a plurality of accurate and relevant attributes”, or “(meta)data [that does not] meet domain-relevant community standards” (Wilkinson et al., 2016, p. 4).
sis platform, for cross-corpus evaluation and analysis. We present the case study of a typological research project on Melanesian languages, where we have successfully extended and used these tools to evaluate and analyse datasets from different sources and in different formats.

2 A software-driven workflow

The workflow we propose bypasses the obstacles of datasets in different formats and missing metadata through a focus on data exploration, and the application of suitable software, i.e., corpus-tools.org. This set of linguistic corpus software includes ANNIS (Krause and Zeldes, 2016), an open source search and visualization platform. ANNIS is a web application implemented with a Java front-end and exchangeable backends; while older and current versions use the object-relational database management system PostgreSQL as backend, future versions will use the faster custom in-memory graph database graphANNIS (Krause, 2019). The software unifies linguistic annotations and structures of corpora in graphs and makes them accessible via its powerful, linguistically-informed query language AQL. Results can be displayed in a wide variety of visualizations, including the ones common for linguistic data, e.g., trees, coreference graphs, etc. Further functionality includes automated frequency analysis based on text and annotation as well as structure and subgraph searches, export of results, and the provision of uniquely identifiable references to queries, result sets and single results via generated hyperlinks. Both text and multimedia corpora are supported, and ANNIS offers playback of video and audio segments. With ANNIS it is possible to conduct cross-corpus data exploration and analysis, independently of original data formats of the corpora.

In order to use the given corpus datasets in ANNIS, they must be converted into a format that ANNIS can import. To this end, corpus-tools.org also includes an open source conversion framework for linguistic data. Pepper (Zipser et al., 2011). During a conversion process with Pepper, data is mapped to an instance of the meta-model Salt (Zipser and Romary, 2010), of which an open source implementation and API in the Java programming language is also a part of corpus-tools.org. Salt is based on a generic graph with semantically sparse layers that concretize the model and API. Following the mapping to this intermediate model, the data is mapped to the target format. The Pepper platform provides the intermediate graph model, an API, a command-line interface, and a plugin mechanism based on OSGi (OSGi Alliance, 2011), a dynamic module system for Java. The mapping process itself is implemented in plugins for import, export, and model manipulation. A Pepper workflow description in XML specifies the order and configuration of plugins to be used during conversion.

The described set of software tools make it possible to implement a workflow for the evaluation of candidate datasets, as well as for the actual analysis in typological and other linguistic studies, via cross-corpus search and visualisation, and iterative cycles of annotation and analysis.

The workflow consists of the following steps, see Figure 1.

1. Compilation of candidate datasets
2. Conversion from source formats to ANNIS format and import in ANNIS
3. Evaluation of candidate datasets’ suitability for the study (query, visualization, analysis)
4. Definition of research datasets based on 3
5. Conversion of research datasets to the format of the annotation software (e.g., MMAX2, EXMARaLDA, GraphAnno, Toolbox, TCF, Tree-tagger, and more)
6. Annotation
7. Conversion from annotation format to ANNIS format and import in ANNIS
8. Analysis via query/visualization in ANNIS
9. Formulation of research results based on 8

Steps 6–8 are usually repeated in iterative cycles of annotation, analysis, adjustment of requirements. One of the main advantages of the proposed workflow is that these steps can be automated by implementing best practices from software engineering: version control and continuous integration (CI, Booch (1992)). The annotated files can be placed in a version control system, which is polled by a CI system.

When changes are committed to version control, the CI system triggers the conversion to

---

2https://www.postgresql.org/
3https://github.com/korpling/ANNIS

---

4Salt differs from LAF (Ide and Romary, 2006) in that: Salt allows annotations on edges; Salt models relations between tokens and base text where GraF (Ide and Suderman, 2007) uses spans for both; in Salt, primary data is part of the model.

5For a list of available modules see http://corpus-tools.org/pepper/knownModules.
the ANNIS format via the Pepper command-line client, and the subsequent import in ANNIS via its REST API. This method also introduces positive side-effects in the workflow: Automated conversion relieves researchers of tedious work; version control allows for unwanted changes to be rolled back; combined with version control, CI introduces reproducibility of annotation-analysis cycles, and enables debugging of erratic processes, and the testability of the automation itself, e.g., by implementing unit and integration tests.

3 Case study: The MelaTAMP project

We have implemented the proposed workflow in the typological research project MelaTAMP.6

The project aims to expand the knowledge about tempus, aspect, modality and polarity systems in mood-prominent languages (cf. Bhat (1999)) through a corpus-based analysis of relevant expressions and contexts in 7 Melanesian languages. To this end, the corpora (see Table 1) have been iteratively annotated for clause types, temporal reference, event structure, modal domain, and polarity. Subsequent conversion and visualization in ANNIS have enabled the analysis of, e.g., expressions of irrealis (von Prince et al., 2018), or habitual contexts (von Prince et al., accepted).

Within the scope of the project, we have started out with a set of seven existing corpora. The

Figure 1: Diagram depicting the proposed workflow based on corpus-tools.org software.

The corpora have been acquired through direct exchange with colleagues, and some had been created by project members (see Table 1). We have also added another set of six corpora to the research datasets, that have been elicited as part of the project, using custom storyboards (von Prince, 2018a,b,c,d,e; Krajinović, 2018a,b,c).

Some of the corpora are available from language archives, but the obstacles to data evaluation presented by the lack of relevant metadata and search options in archive interfaces, mentioned in the introduction, pertain, and we have bypassed them by acquiring the respective datasets directly from their authors. Nevertheless, the dataset candidates for our different research questions have been evaluated and selected as intended.

As most of the corpora were available in the Toolbox format, we chose to annotate directly in the Toolbox text format files with the help of a text editor (Sublime Text 37).

We have automated the annotation-analysis cycle by versioning our Toolbox text format files with Git (Chacon and Straub, 2014) on our institutional GitLab8 instance. GitLab’s CI system has been configured to poll the repository holding the annotations, and run a script on a virtual machine whenever files have changed. The script installs Pepper, installs the necessary conversion plugins,

6https://hu.berlin/melatamp

7https://www.sublimetext.com/

8https://about.gitlab.com/
Table 1: Overview of corpora used in the MelaTAMP project. Bold language names signify that a corpus has been elicited during the project. Software: “Toolbox” = (unknown) version of SIL Toolbox (Robinson et al., 2007); “ELAN” = (unknown) version of ELAN (Wittenburg et al., 2006); “FLEx” = (unknown) version of SIL FieldWorks Language Explorer (https://github.com/sillsdev/FieldWorks).

<table>
<thead>
<tr>
<th>Language</th>
<th>ISO 639-3</th>
<th>Tokens</th>
<th>Country</th>
<th>Elicitator</th>
<th>Format (Software)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daakie</td>
<td>ptv</td>
<td>~86k</td>
<td>Vanuatu</td>
<td>Krifka (2013)</td>
<td>Text (Toolbox)</td>
</tr>
<tr>
<td>Daakie</td>
<td>ptv</td>
<td>~3k</td>
<td>Vanuatu</td>
<td>Manfred Krifa</td>
<td>Text (Toolbox)</td>
</tr>
<tr>
<td>Daakaka</td>
<td>bpa</td>
<td>~59k</td>
<td>Vanuatu</td>
<td>von Prince (2013a)</td>
<td>Text (Toolbox)</td>
</tr>
<tr>
<td>Daakaka</td>
<td>bpa</td>
<td>~80k</td>
<td>Vanuatu</td>
<td>Kilu von Prince</td>
<td>XML (ELAN)</td>
</tr>
<tr>
<td>Dalkalaen</td>
<td>bpa</td>
<td>~30k</td>
<td>Vanuatu</td>
<td>von Prince (2013b)</td>
<td>Text (Toolbox)</td>
</tr>
<tr>
<td>Dalkalaen</td>
<td></td>
<td>~13k</td>
<td>Vanuatu</td>
<td>Kilu von Prince</td>
<td>XML (ELAN)</td>
</tr>
<tr>
<td>North Ambrym</td>
<td>mmg</td>
<td>~24k</td>
<td>Vanuatu</td>
<td>Franjieh (2013)</td>
<td>XML (FLEx)</td>
</tr>
<tr>
<td>North Ambrym</td>
<td>mmg</td>
<td>~15k</td>
<td>Vanuatu</td>
<td>Michael Franjieh</td>
<td>XML (ELAN)</td>
</tr>
<tr>
<td>Mavea</td>
<td>mkv</td>
<td>~30k</td>
<td>Vanuatu</td>
<td>Guérin (2006)</td>
<td>Text (Toolbox)</td>
</tr>
<tr>
<td>Mavea</td>
<td>mkv</td>
<td>~12k</td>
<td>Vanuatu</td>
<td>Valérie Guérin</td>
<td>Text (Toolbox)</td>
</tr>
<tr>
<td>South Efate</td>
<td>erk</td>
<td>~54k</td>
<td>Vanuatu</td>
<td>Thieberger (2006)</td>
<td>Text (Toolbox)</td>
</tr>
<tr>
<td>South Efate</td>
<td>erk</td>
<td>~15k</td>
<td>Vanuatu</td>
<td>Ana Krajnovic</td>
<td>XML (ELAN)</td>
</tr>
<tr>
<td>Saliba/Logea</td>
<td>sbe</td>
<td>~138k</td>
<td>Papua</td>
<td>Margetts et al. (2017)</td>
<td>Text (Toolbox)</td>
</tr>
</tbody>
</table>

The import plugin is passed a Toolbox text format file or a directory containing such files. It will parse the files, validate them, and transform their contents into a Salt graph structure of corpora and documents. Documents have their own document graph, which contains the language data as nodes and edges. It will contain two base text nodes, whose text values represent:

- the lexical information from lines in the Toolbox file marked with `\tx`;
- the morphological information from lines marked with `\mb`.

Token nodes segment the text base according to Toolbox’ interlinearization; Salt can handle Toolbox’ double segmentation by aligning tokens through a sequential data structure. The plugin also detects invalid interlinearizations in the Toolbox data based on incongruencies over token indices, and records them.

Token nodes can have multiple annotations – annotations are realized in the graph model as labels on nodes – and can be covered by span nodes which in turn can have multiple annotations themselves. Span nodes are used to represent `\ref` phrases and `\id` documents.

The import plugin can be passed parameters to configure the conversion. These include: specifications for marker distinction; how morphology delimiters should be handled; whether markers should be changed during conversion; if and how interlinearization errors should be recorded, and

---

9https://github.com/korpling/pepperModules-ANNISModules
10https://github.com/korpling/pepperModules-ElanModules

converts the annotation data into the ANNIS format, and uploads the converted files to an ANNIS instance via REST API.

Following the workflow, we needed the following Pepper plugins:

- Toolbox text format import plugin
- FLEx XML import plugin
- ELAN import plugin
- Toolbox text format export plugin
- ANNIS format export plugin

An export plugin for the ANNIS format already exists, as does an ELAN import plugin. However, the ELAN import plugin has been developed for a relatively narrow set of use cases, and did not yield usable results for our case. Instead, we exported the corpus files that were available in the ELAN XML format to the Toolbox text format via the respective export functionality in ELAN.

In the course of the project, we have developed three further plugins in two open source software projects: ToolboxTextModules and FLExModules.

3.1 ToolboxTextModules

The ToolboxTextModules (Druskat, 2018b) project holds a Pepper import plugin to map the Toolbox text format to a Salt model, and an export plugin to map a Salt model to the Toolbox text format.
more. Additionally, the plugin supports a custom structure which cannot be expressed in the Toolbox software, but is supported by the format and leveraged by our annotation process: Annotations in Toolbox can only be assigned to either lexical or morphological units, or to the whole $\text{ref}$ span. We have introduced sub-phrases termed $\text{subref}$, which are index-determined spans which cover a subset of the complete set of morphological token nodes within a Toolbox $\text{ref}$ unit. These can be used to annotate, e.g., clauses.

The export plugin converts a Salt model to Toolbox text format files, and adheres as closely as possible to the Multi-Dictionary Formatter specifications (Coward and Grimes, 2003). Parameters are again used to configure the conversion, mainly to specify which annotation layers in the Salt model should be mapped to which markers.

### 3.2 FLExModules

FLExModules (Druskat, 2018a) provides an import plugin for the XML format exported from FieldWorks Language Explorer (FLEx). It transforms the interlinearized text structure provided in the XML to a Salt model, complete with corpus structure and annotations. This model is very similar to the model produced during conversion from the Toolbox text format, but the implementation is less complex than for the import plugin for the latter, given the structured nature of the data. The FLEx XML import plugin provides parametrization inasmuch as annotation namespaces and names can be changed in the process using a pre-defined mapping.

### 4 Conclusion

In the MelaTAMP research project we have successfully implemented and automated the annotation-analysis part of the proposed workflow (see Figure 1) for 13 language documentation corpora provided in three different data formats, and have enabled future implementations of the evaluation part by creating ToolboxTextModules and FLExModules. In this paper we have also shown that a workflow driven by extended corpus-tools.org software can bypass obstacles to data evaluation presented by insufficient implementation of the FAIR principles for data management in language documentation datasets as available in, e.g., archives of endangered languages. The presented workflow features automation as a means to boost efficiency and reduce errors. It enabled us to analyse expressions of irrealis and habitual contexts across 7 Melanesian languages, and subsequently formulate research results such as von Prince et al. (2018) and von Prince et al. (accepted).

### 5 Acknowledgments

We would like to thank Thomas Krause, the project lead for ANNIS, for his continued support. The project “A corpus-based contrastive study tense, aspect, modality and polarity (TAMP) in Austronesian languages of Melanesia (MelaTAMP)” has been funded by Deutsche Forschungsgemeinschaft (DFG) under grant no. 273640553.

### References


Valérie Guérin. 2006. Documentation of Mavea. SOAS, Endangered Languages

Nancy Ide and Keith Suderman. 2007. GrAF: A Graph-Based Format for Linguistic Annotations.


Manfred Krifka. 2013. Daakie, The Language Archive. MPI for Psycholinguistics. https://hdl.handle.net/1839/00-0000-0000-000F-4E20-B@view, Nijmegen.


Kilu von Prince. 2013a. Daakaka, The Language Archive. MPI for Psycholinguistics. https://hdl.handle.net/1839/00-0000-0000-000F-4E20-B@view, Nijmegen.

Kilu von Prince. 2013b. Dalkalaen, The Language Archive. MPI for Psycholinguistics. https://hdl.handle.net/1839/00-0000-0000-000F-4E20-B@view, Nijmegen.


Kilu von Prince, Ana Krajinović, Anna Margetts, Valérie Guérin, and Nick Thieberger. accepted. Habituality in four Oceanic languages of Melanesia. Language Typology and Universals.


