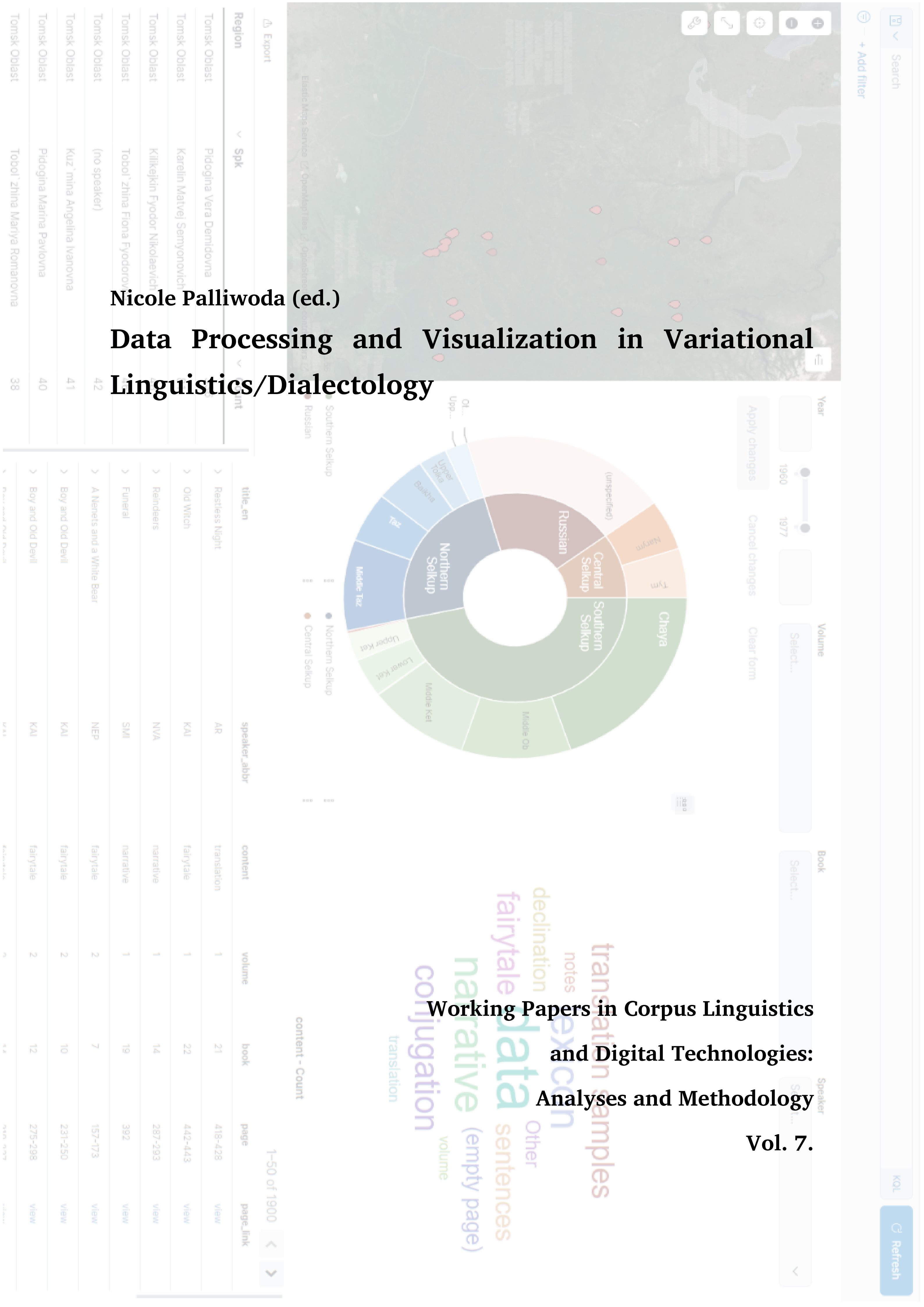


Nicole Palliwoda (ed.)

Data Processing and Visualization in Variational Linguistics/Dialectology

Working Papers in Corpus Linguistics and Digital Technologies: Analyses and Methodology

Vol. 7.

Nicole Palliwoda (ed.)

Data Processing and Visualization in Variational Linguistics/Dialectology

Working Papers in Corpus Linguistics and Digital Technologies:

Analyses and Methodology

Vol. 7.

Szeged – Hamburg

2022

Working Papers in Corpus Linguistics and Digital Technologies: Analyses and methodology Vol.7

WPCL issues do not appear according to strict schedule.

Vol. 7 (2022)

Published by
University of Szeged, Department of Finno-Ugric Studies
Egyetem utca 2. 6722 Szeged
Institut für Finnougristik/ Uralistik
Überseering 35, 22297 Hamburg

Published 2022

ISBN 978-963-306-872-4 (pdf)

Preface

From 3 to 4 September 2020, the workshop *Data Processing and Visualization in Variational Linguistics/Dialectology* took place as a digital event organized by Ludwig Maximilian Breuer (University of Vienna), Kai-Uwe Carstensen (University of Siegen) and Nicole Palliwoda (University of Siegen) via the University of Siegen. This workshop is a continuation and extension of the very successful and constructive workshops that took place in 2016 and 2017. The event *Exchange: Data Preparation and Management of the DMW Project* organized by Nicole Palliwoda (University of Siegen) kicked off on August 30, 2016 at the University of Siegen. The focus was on direct exchange with various colleagues (including Ludwig Maximilian Breuer, University of Vienna; Christoph Draxler, Ludwig-Maximilians-Universität Munich; Timm Lehmberg, University of Hamburg) from different projects (e.g. *Deutsch in Österreich/DiÖ*; *Indigenous Northern Eurasian Languages/INEL*; *Siegerländer Sprachatlas/SiSAL*) and institutions, particularly on the topics of purpose, usefulness and connectivity of different (transcription) tools and databases, with a view to the start of the academy project *Dialektatlas Mittleres Westdeutschland (DMW)* in July 2016, which aims to create an audiovisual dialect atlas that will include the currently still graspable dialects and varieties of German. The aim of the project is to create an audio-visual dialect atlas that will document the dialects and varieties in North Rhine-Westphalia that can still be grasped and make them accessible on maps to the scientific community and the interested public.

On 7 July 2017, the second workshop *Linguistic Data and Databases (LDDB)* took place at the University of Vienna, organized by Ludwig Maximilian Breuer. The focus was on the digital processing and sustainable preparation of complex data (natural language) using various databases (including SQL and graph databases, online tools), but also on fundamental questions regarding the structuring and implementation of surveys, collaborative work (including experiment software, project management tools) and topics relating to citizen science, open data and open science. In addition to the participants who had already been recruited in 2016, Robert Engsterhold (University of Marburg, *regionalsprache.de/REDE*) and Christoph Purschke (University of Luxembourg, *Lingscape*) also took part.

Furthermore, Kai-Uwe Carstensen (University of Siegen), developer of the digital DMW atlas, presented first sketches of the dynamic DMW preview word maps, which have been published by now, and are online available as described in his contribution in this volume.

At the third workshop in September 2020, the focus of the workshop was on the post-processing or preparation of linguistic data and on digital tools. This means that in particular – but not only – visualization was in the foreground. Visualization is understood to mean, on the one hand, the actual presentation of data in the form of (digital) maps, diagrams and the like. On the other hand, however, it also includes the design of user interfaces of the digital linguistic tools. Accordingly, in the workshop, various technical means for implementing such projects were presented and discussed directly from practice, i.e. on the basis of experience from international (variation) linguistic research projects. A special focus of the event was and still is the exchange between the individual projects and participants, which is reflected in a presentation time of 20 and a discussion time of 40 minutes per project/speaker.

The publication now brings together a total of five contributions that provide insights, results, problems and reflections on different methods and projects from the workshop.

Nicole Palliwoda (Kiel)

Contents

DOI: https://doi.org/10.14232/wpcl.2022.7.1

Transcribe: a Web-Based Linguistic Transcription Tool

Ludwig Maximilian Breuer, Arnold Graf, Tahel Singer and Markus Pluschkovits

1. Introduction and Overview

Transcribe is a transcription application currently being developed in the context of the task-cluster E of the Special Research Programme (FWF F 60) "German in Austria. Variation – Contact – Perception" (abb. as SFB DiOE in the following). The present paper aims to describe the principles of its development in the context of the SFB DiOE, its solutions for audio processing and visualization, and its use-case as a transcription tool for linguistics in general and linguistic research groups in particular. We conclude with a practical application of automatic translation of eye-dialect transcripts of Viennese to orthographic standard German transcripts, utilizing the automatic tokenization and multiple tiers of Transcribe.

Transcribe was developed as an offline-first web app. The main advantages of web-based applications are obvious: web applications are compatible with all operating systems that are able to run modern browsers, and thereby eliminate the issue of porting software. The distribution of web-based applications is extremely simple in comparison to traditional desktop software – instead of going through an installation process, the users simply navigate to a URL in their browser, which can easily be bookmarked. Updates on the app are transparent and do not require any action from the user.

Transcribe was developed with linguistic research groups in mind, and with the goal to support and enhance their workflow with a variety of functions. In contrast to many other common applications, Transcribe utilizes a client-server-architecture, which enables a link to a central back-end. This central back-end can host all the relevant (linguistic) research data, as is the case in the SFB DiOE (see DiOE 2020: DioeDB).

However, Transcribe is also designed to support smaller-scale projects, where a centralized server storing the research data is either unfeasible or unnecessary. For this reason (and to increase resilience against high latency and network issues), Transcribe was conceived as an offline-first application (see Linklater et al. 2018: 260), meaning that it can also be used locally, while still retaining almost all of its functions. Transcribe can therefore either be used as a stand-alone tool, or as a part of a toolkit.

Some of the basic tenants of Transcribe can be summarized by the concepts of user-centered design, collaboration, and flexibility. The goal of the development is to combine the smooth interactivity of modern, native desktop apps with the benefits of browser-based web applications. To facilitate this, some technical improvements had to be made, which shall be showcased in the following paragraphs.

The development of Transcribe is recorded in a public GitHub repository (see DiOE 2020: Transcribe). The aim is to disseminate the code as widely as possible and adhere to basic concepts of the Open-Science movement (see European Commission 2016: 33).

Nicole Palliwoda (ed.): *Data Processing and Visualization in Variational Linguistics/Dialectology*. Working Papers in Corpus Linguistics and Digital Technologies: Analyses and methodology. Vol.7. 2022.

2. User Interface

2.1. Principles of Transcribe's UI-Design

The intended use of Transcribe is within both a professional and scientific setting, the userbase is therefore conceived as any such people that are concerned with linguistic data in these contexts. They use the tool repeatedly, for longer periods of time, and utilize its functions parallel or sequential during different steps of their workflow. For this reason, the goalpost for Transcribe's UI was what has been termed the sophisticated user (see Debasmita and Ardhendu 2015: 130) and their requirements. In the spirit of user-centered design, the development process has been accompanied by periodic meetings between the developers and users of Transcribe in order to identify issues and improve the UI iteratively (Chammas, Quaresma and Mont'Alvão 2015: 5398).

Despite the orientation towards sophisticated users, Transcribe aims to provide an intuitive user interface for first-timers. This was enabled by following some core tenets of the philosophy of Interaction Design (see Debasmita and Ardhendu: 131–134 and Nielsen and Molich 1990: 251pp). Among them are the following:

- The principle of familiarity (see Raskin 1994: 17): If possible, Transcribe does not introduce new or formerly unknown terms, icons, or means of interacting with the program. Transcribe relies on established shortcuts, either known from other transcription software (e.g. Exmaralda), or familiar from the operating systems of current desktop computers. For example, Transcribe enables users to select multiple elements by keeping the shift or control key pressed, it offers context-sensitive menus, and supports pinch-to-zoom to zoom in or out of the waveform-visualization, which are all features users usually know from their operating system.
- The principle of discoverability: The bulk of Transcribe's features can be found in the submenu actions, which also highlights their respective keyboard shortcuts. More specialized functions appear automatically in those contexts where they are actually needed – e.g. a pop-up on-screen keyboard for IPA characters in the tier dedicated for phonetic transcription, or in the search bar.
- The principle of consistency (see Nielsen 1999: 2): Visual elements which resemble each other should have similar functions and support similar interactions. A coherent design scheme, as was utilized for Transcribe, provides an implicit system of rules of interaction for both the users and the designers.
- The principle of safety of interaction: in order to facilitate fluid interaction with an application, especially those that are concerned with the entry of data, mechanisms that prevent user errors (or enable the users to undo them painlessly) must be put in place. The application should be what is called a "relaxed environment," which does not require constant, intense focus (Debasmita and Ardhendu 2015: 132). To help create such a relaxed environment, Transcribe uses a generous undo-redo-system, locally saving up to 1500 individual operations, displaying them visually in a dedicated history, and allowing for selective undoing of them. Undoing and redoing individual operations can be done by using the standard keyboard shortcut of the end-user's device, which hopefully makes users intuitively default to them. Additionally, Transcribe periodically saves the latest version of a current transcript in a local cache as long as the transcript is being worked on, which means that it can be recovered even after a system crash.

2.2. Organization and Implementation of the UI

As in almost all subdisciplines of software development, developing a graphical UI employs the principle of isolation of functional units (cf. Krasner/Pope 1988: 26). While the benefits of this approach can be assumed to be widely known, the following summarizes them succinctly:

> Isolating functional units from each other as much as possible makes it easier for the application designer to understand and modify each particular unit (Krasner and Pope 1988: 26).

The UI of Transcribe is being developed with the Vue.js framework in accordance with the principles stated above. For most of its interactive control elements, it employs John Leider's Vuetify (see Leider 2020) as user interface library.

Vue.js is a declarative framework for the creation of UIs for web applications. In contrast to the more traditional Model-View-Controller (MVC) architecture, Vue.js implements the more modern concept of the Model-View-ViewModel (MVVM), which allows for the binding of individual parts of the UI directly to the data model. If the data model is changed, Vue.js identifies necessary updates on the level of the graphic UI (GUI) and applies them (see You 2020: Vue.js Introduction). This mechanism takes work-load off of the developers and reduces the risk of possible discrepancies between data model and the interface of the program. A drawback of this type of architecture is its comparatively high memory consumption in contrast to the MVC-architecture (see Gossman 2006).

Like other libraries and frameworks, Vue.js fosters the development of modular, component-oriented applications by employing single-file-components, i.e. small, self-contained program parts. These component parts usually have very limited, but specific functions – e.g. portrayal of a menu or a text box – and define fixed interfaces to communicate with other components. Combining these short, manageable components creates a component tree, which serves as GUI.

To further the development of Transcribe, an additional library of often-used standard components was employed as well. Vuetify (see Leider 2020: Vuetify) utilizes Material Design, a design system launched by Google (Google 2014: Material Design), and contains several so-called Widgets. While Transcribe is not completely bound to the rules and principles of Material Design, it orients itself on them. As a result, the inherent coherence of the rules and requirements of the Material Design framework, implemented in Vuetify, helps to facilitate the visual and functional consistency of Transcribe.

3. Audio Processing and Visualization in Transcribe

Initially, three important requirements for the program were identified:

(1) A speed and response-time comparable to that of a native desktop application
(2) The implementation as a web-based application
(3) The possibility of utilizing the application without a server back-end.

These requirements result in the need for an efficient, client-based method of decoding, visualizing and analyzing audio data. This, unfortunately, excludes traditional de-facto standards, as these are usually oriented towards server-side execution of these tasks, or in the context of native applications (see e.g. FFmpeg, FFmpeg 2016). The following describes issues which were faced during the fulfillment of these requirements, and their implemented or potential solutions.

3.1. Segmentation of Compressed Audio Data

The Web Audio API was specified in 2011 by the World Wide Web Consortium (W3C) (see W3C 201: WebAudio). It allows for the decoding of audio data in different formats. 'Decoding' in this context means the approximation (or, somewhat fuzzier: the 'tracing back') of compressed data to the originally digitized, time-discreet sequence of pulse-code modulated (PCM) signals. The individual signals depict the amplitude of a given sound wave in a specific point in time as samples, and are therefore suited for visualization and analysis of audio signals (in contrast to the compressed format). The decoding is obviously also a necessary step for playing the audio file, for which individual sampling points are converted into electrical voltage by means of a digital-analogue-converter, which powers the membrane of speakers, and thereby making it audible.

The current (Februar 2020) implementations of the decoder for ogg/vorbis in the popular browsers Chrome and Firefox do not allow to decode compressed audio streams continuously or in pre-defined chunks. After the handover of the audio-buffer to the decoder, the decoder will decode the audio in its entirety before handing it back to the client. The ratio of the duration of the audio material to the duration of its decoding is approximately 30:1, i.e. decoding about 30 minutes of audio material takes roughly one minute, becoming more balanced (in this case: slower) the longer the audio is.[1] During the decoding process, a considerable strain is put on the client, which limits other functions of the application. Other JavaScript-based decoders, such as Audiocogs (see Audiocogs 2015), have been considered, but ultimately did not bring a sufficient performance enhancement in comparison to the WebAudio API. In the context of Transcribe, and the SFB:DiOE, which deals with recordings of up to (sometimes over) two hours, it was vital to find a more efficient solution.

To solve this problem, an ogg-bitstream-parser and -chunker was formulated in Typescript, based on the specification of the RFC 3533 of Silva Pfeiffer (2003). These are able to read and chunk the binary format of ogg audio pages and containers (see table 1).

[1] These are reference values at best, established by internal tests. The tests were conducted on a standard MacBook Pro with an Intel Skylake i5 CPU@3,1GHz, without dedicated hardware for decoding, on both Firefox and Chrome.

Table 1 *Schematic representation of an .ogg page in binary*

Bit 0-7	Bit 8-15	Bit 16-32	Bit 24-31	Byte
Magic Number (Marker) for the beginning of „OggS“ in ASCII				0-3
Ogg-version	Header-type			4-7
Starting point as a 64-bit integer in miliseconds				8-11
		Serial number of		12-15
the bit stream		page		16-19
sequence		check		20-23
sum		number of segments		24-27
Vorbis-encoded audio segments				28–…

This allows to index and correctly chunk not fully loaded and still compressed ogg-audio files. These chunks are identified during the loading of the binary blob and can be handed to the decoder piece by piece. The decoded results can be handed piecewise as well to the functions responsible for visualization and analysis of the audio sample without delaying the users significantly during the process. Furthermore, this enables loading individual segments of an audio file in advance by HTTP-Range-Requests (see Mozilla Developer Network 2021: Range Requests), meaning that users can skip to a later part of the transcript without waiting for the complete audio file to be loaded and decoded.

This parsing algorithm has a run-time complexity of *O(n)* and is optimized in such a way that for the use cases described, it can deliver results in the range of single-digit milliseconds. As of Feburary 2020, it is currently the only stand-alone implementation of such an algorithm in JavaScript, and is gonna be published as a library in the GitHub repository of the SFB DiOE.

3.2. Waveform Visualization

The term ‘waveform’ or ‘oscillogram’ means the visual representation of the envelope of acoustic waves in a diagram and is one of the most common graphical representations of audio data. In the specific application of transcribing spoken conversations, the waveform can only marginally give information

about acoustic phenomena such as pitch or articulation, however, it does aid with identifying pauses and differentiating between multiple speakers (see figure 1).[2]

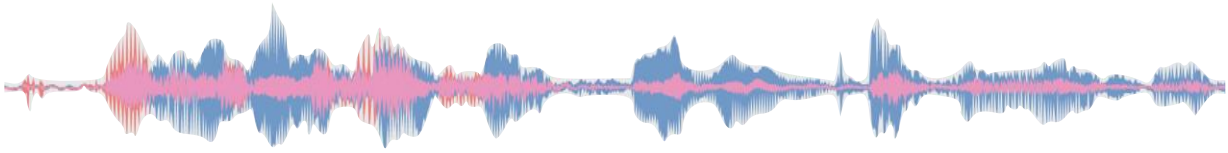


Figure 1 *Representation of a two-channel waveform in Transcribe*

As mentioned above, Transcribe operates within specific constraints that often exclude traditional server-sided solutions and require a great deal of efficiency of the algorithms used. After the evaluation of existing, client-based program libraries[3], a new solution was developed, with properties suitable for the task at hand.

For the output format of the waveforms, scalable vector graphics (SVG) were choosen, as these can be scaled without a loss of quality, in contrast to raster graphics. This furthermore allows for zooming in and out of the visualization without requiring a new computation of it. Additionally, both high- and low resolution screens can utilize the same visualization without any loss in either efficiency or quality. This obviously applies to all vector graphics.

The present solutions generates the complete SVG waveform once as an XML-string, and – in contrast to other solutions – not in iterations as a tree-like structure of Document-Object-Model elements (DOM-objects). The only access of the DOM during the execution is the embedding of the graphic in each segment of the document, which corresponds to the parsing of the structure as DOM-element. In this way, the present solution requires much less memory, resulting in a ten to 15 times shorter run-time than comparable DOM-based solutions (cf. Justice 2014). Because this solution avoids accessing the DOM-interface, it allows utilizing the algorithm in contexts which usually do not have access to the rendering engine of a browser – e.g. by using Node.js on servers, or aside of the main threads in Web Workers (which is the case for Transcribe).

Another novelty of this algorithm is that it allows for the simultaneous processing of two channels of a recording at once. Because two audio streams of one recording are necessarily of equal length, the same loop can be used to generate several independent wave forms. While this optimization seems trivial, it has not been employed by any of the libraries surveyed, and it saves about 35% of run-time.

The implementation presented here is, as of February 2020 and to our best knowledge, the fastest JavaScript-based library for the creation of SVG waveforms. Its optimizations result in a run time of about 15-20 miliseconds for a two-channel audio stream of 33 seconds of length, with a sample rate of 150 points per second. This corresponds to a ratio of length of audio stream to run time of about 1:2200. The implementation can be found as a library on GitHub in the repositories of Deutsch in Österreich.

[2] This is the case for stereo recordings, where each channel can be assigned to a speaker. This method of recording has proven itself to aid understanding of the transcriptors in the case of the SFB.

[3] Specifically DrawWave (https://github.com/meandavejustice/draw-wave), WaveSurfer (https://github.com/katspaugh/wavesurfer.js), and Audio-Waveform-SVG-Path (https://www.npmjs.com/package/audio-waveform-svg-path)

3.3. Spectrogram Visualization

As has been mentioned before, the waveform visualization of audio may serve as a signpost assisting transcriptors during longer stretches of conversation. It is, however, not suited for linguistic analysis proper, which may place its focus on acoustic and phonetic characteristics. The overlay of multiple signals or waves of different frequencies (caused by e.g. the overtones of a vowel sound) creates sometimes barely parseable visual representations. A spectrogram is an alternative visual representation of an audio signal, which not only visualizes a signals amplitude, but the whole spectrum of frequencies on a time axis. The spectrogram is therefore based on the transformation of the domain of time of a signal to the domain of its frequencies. Spectrograms are usually used for phonetic research.

The aforementioned conversion to the domain of frequency is achieved through the so-called Fourier-transformation. The algorithm on which the most common implementation of this method is based was originally described by Carl Friedrich Gauß[4], which was rediscovered by Cooley and Tukey in 1965 (see Heideman et al. 1985: 265p), and is now known as the Fast-Fourier-Transformation (FFT).

Because of Transcribe's plug-in architecture, the spectrogram can – just like the wave forms – be selected as standard mode of visualization, and is completely scroll-able. In order to make this achievable without excessive loading times, the FFT implementation has to be as efficient as possible. The implementation currently used by Transcribe is an adaption of the method included in the signal processing library DSP.js (see Brooks 2017). It has been chosen due to its array of window functions, which makes it suitable for a variety of recording situations and levels of audio quality (see National Instruments 2019).

The current method is able to transform about 30 seconds of audio material in roughly 500-700 milliseconds, and visualize it as a spectrogram. While this is fast enough to be suitable for analysis, it may necessitate short waiting times while navigating in a longer transcript. The optimization of this process, and therefore, of the user experience (see Nielsen 2012) of Transcribe, is still a subject of development. The following approaches have been tried or considered;

(1) Expanding the FFT-package by Fødor Indutny with multiple window functions, which can, according to the benchmarks, achieve better perfomance under certain conditions (see Indutny 2017 and Audioplastic 2017).
(2) The implementation of the same algorithm in the WebAssembly run-time environment. In an initial trial run, the method mentioned above was ported into AssemblyScript and compiled in WebAssembly. This, however, did not result in a significant increase in performance. The trial run and tests are archieved on the GitHub repositories of Deutsch in Österreich for purposes of reproduction (cf. DiOE 2020: Transcribe).
(3) Parallelizing the FFT, or more specifically, the execution of the FFT via the WebGL-API specified by the Khronos Group (2020). This utilizes the possibility of expressing the Fourier transformation as multiplication of matrices. The viability of this approach could be proven outside of the Browser environment (see Rosenberg 2018), and Google's Machine-Learning Library TensorFlow offers a basic implementation for JavaScript environments (see Google 2019). The advantages of this method are especially pronounced with large amounts of data (see Demorest 2007 and Sasiki 2020).

[4] The original manuscript was unpublished. Heideman et. al. (1985: 266) date the writing of this manuscript around 1805.

The method suggested in (3) seems, at this point, the most viable route to pursue. At the same time, it also represents the largest deviation from the typical path of generating spectrograms.

4. Transcription with Transcribe

After having considered some of the more technical aspects, the following chapter turns towards the linguistic application of Transcribe. This includes a discussion of transcriptions and transcription conventions, especially in the context of a large-scale research project, and a possible application of Transcribe for machine translation, based on an N-gram approach as done by Tahel Singer. As mentioned above, Transcribe was developed in the context of the SFB DiOE, a large-scale variationist research program researching the variation, contact and perception of different varieties of German in Austria (for an overview of the research program, see Lenz 2018 or DiOE 2018). As the research program is situated in five different institutions, with nine different project parts, the research foci of the individual project parts are diverse – as are the requirements for the transcription of spoken language data.

Transcribe tackles this issue on two different fronts: on the one hand, issues stemming from multiple researchers and assistants working on the same transcripts are avoided by Transcribe constantly updating the changes made to the transcript, and providing a history of changes made. Transcribe utilizes the internal data base of the research project as back-end, and changes made by one user on a specific transcript are immediately updated and displayed for other users, avoiding issues of versioning local transcript data (see DiOE 2020: Transcribe). This means that researchers across the different institutions always work with the same, up-to-date transcript. On the other hand, Transcribe allows different transcription conventions, and parses them accordingly. Therefore, discourse-oriented project parts can transcribe their data according to the GAT2 standard, while other project parts can utilize eye-dialect or orthographic transcriptions.

While it may seem counterintuitive to employ several transcription conventions within one research program, the research matter at hand necessitates this. Obviously, there is no singular ‘correct’ transcription standard, and the transcription convention chosen ultimately needs to be suited for the concrete research at hand (cf. Nagy and Sharma 2013: 242). This is due to a variety of reasons, among them being the desired level of detail of the transcription, issues of searchability, or the feasibility of transcribing larger amounts of data. For this reason, several different transcription systems are employed throughout the SFB DiOE, among them close phonetic transcription with IPA symbols, eye-dialect and standardized orthographic transcripts, and transcripts modelled after the GAT2 standard. The choice of transcription convention used reflects the research interest of the individual project parts. But, as Kendall (2008: 337) cautions, the act of transcribing spoken data is far from theory neutral, and influences possible further analyses of the data. For this reason, Transcribe offers multiple tiers for transcription, which enables researchers to transcribe the same token phonetically, orthographically, or with an eye-dialect system. In such a way, transcribing audio data on different tiers can create parallel, time-aligned corpuses.

A further danger identified by Nagy and Sharma (2013: 242), which especially endangers the reusability of transcripts, concerns the usage of punctuation in transcription and the ambiguity this can create. While there are some (competing) standardized protocols in place which formalize the meaning of punctuation, this issue is greatly elevated by supplying the audio recordings to the transcript via time-alignment. Additionally, Transcribe allows for customizable type-token parsing, meaning that

certain punctuation conventions can be coded into the transcript, and become meaningful to the software, e.g. indicating contractions with underscores, which changes the token type of the respective tokens. This type-token parser can be customized using regular expressions.

As a result of these features of Transcribe, especially the possibility of transcribing the same token on different tiers, and the modular programming of Transcribe, a further use-case is introduced in the following – the application of Transcribe in machine translation. This translation task between a transcription tier featuring an eye-dialect transcript of Viennese to a tier of a standard-orthographic Standard German following an N-Gram approach was developed as part of T. Singer's bachelor's thesis at the Technical University of Vienna. No deep learning features are involved in the main translation task, following the assumption that supervised machine learning methods, i.e., a single layer of local memory network, would be sufficient for this translation task and provide satisfactory results. An additional focus is an experimental approach toward handling unknown words, also known as out-of-vocabulary (OOV) words, involving a heuristic method and a prediction task of an external pretrained model provided by BERT ('Bidirectional Encoder Representations from Transformers'). The linguistic value of such a task is obvious: by virtue of such a model, it would become possible to automatically add a tier of orthographic standard transcription to an existing eye-dialect transcript. This does not only greatly enhance the searchability of a given transcript, it also allows for the application of other resources, such as automatic Part-of-Speech-tagger, which usually achieve the best results when confronted with the orthographic standard of a given language.

4.1. A Possible Application of Transcribe – Machine Translation Viennese to Standard German

The following therefore addresses the need for machine translation capable of handling language varieties that are not standardized and mostly suffer from lack of natural language processing (NLP) resources. The translation of Viennese from its dialectal form to its orthographic standard, which tends to be identical to Standard German, is the intended goal, while preserving language phenomena that can be found in slightly different grammar rules, idioms and ways of expression.

The data at the core of this work does not stem from the SFB DiOE proper, but rather L. M. Breuer's dissertation project (Breuer 2021), in cooperation with the project part 11 of the SFB DiOE at the Centre for Translation Studies at the University of Vienna. The corpus documents the variety of the German language and coexisting varieties of speech in Vienna. The data is constructed in such a way that the source language does not have any capitalized words and the target language includes capitalized words whenever it is necessary (e.g. proper nouns).

4.2 Approaches and Methods

The ambiguity of individual words characterizing the source language poses a great challenge for this particular translation task and influences the complexity of the translation task (see Trost 2016). The work combines different methods to achieve an optimal translation output; the main translation task is accomplished by observing the context of the word, following the N-gram approach. This approach is in accordance with the principle that single words may not be the best atomic units for translation, especially in this specific case, where the phenomenon of ambiguity is common and can be resolved only by considering a wider context (see Koehn 2009: 127–154).

Modeling with N-grams includes the probability distribution of tuples of the size N that construct word sequences. A unigram represents the sequence of single words, bigrams the sequences of pairs and trigrams the sequences of three consecutive words and they resemble different degrees of translation units. Each gram is assigned a probability using the chain rule probability of the following scheme:

$$P(w_1 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \dots P(w_n|w_{1:n-1}) = \prod_{k=1}^{n} P(w_k|w_{1:k-1})$$

where:

$P(w_1 \dots w_n)$: the probability of a sentence word$_1$, word$_2$, … word$_n$.

It is given by the multiplication of the conditional probabilities;

$P(w_1)$: the probability of the first word;

$P(w_2|w_1)$: the probability of w_2 to appear based on the knowledge that w_1 appeared beforehand; and

$P(w_3|w_{1:2})$: the probability of w_3 to appear based on the knowledge that the sequence w_1 w_2 came before (cf. Koehn 2009: 181–216).

For handling the out-of-vocabulary words, a special mechanism based on known language patterns of sound shifting in Viennese was developed, applying a deeper character-based analysis.

The machine translation consists of the following statistical components; a language model based on trigrams of the target language, a phrase table consisting of uni-, bi- and trigrams of the translation units and their corresponding statistics and a stack decoder. The phrase table enables a deeper resolution of the single words or pair of words for seeking better and more precise translation alternatives. The stack decoder maintains the decoding procedure efficiently; once the input sentence is segmented into phrases, the output sentence is built sequentially from left to right, creating multiple hypotheses that can be referred to as the translation options. The stack manages the hypothesis expansion, i.e., the build-up of the translated sentence, and each hypothesis with the same number of words translated is placed in the same stack (see Koehn 2009: 155–180).

4.3 Handling Out-Of-Vocabulary Words

The default stack decoder does not support a partial translation in case of unknown words, also known as out-of-vocabulary (OOV) words and returns an empty value. With unknown words, we refer to tokens that are not contained in the corpus. The bigger the corpus is, the fewer words are found to be unknown to the system during the lookup process. For this reason, a pre-sentence analysis (also known as the preprocessing step) for the detection of such words is needed and can be easily done by a lookup function in the unigram-based dictionary. The unknown words are divided into two categories: Words that appear neither in the source language nor in the target language and words that do not appear in the source language, but do belong to the target language. The second category exists due to the intelligibility of Viennese and standard German and the switching phenomenon.

4.3.1 Prediction Process

Detected in-target words are added as their translation to the lookup dictionary, i.e., the function is described by the mapping f(w) = w, and the phrase table with the assigned probability value 1.0. This prevents the phrase table's error key from leading to an empty return value and enables the dynamic

enlargement of the corpus giving it "learning" qualities that spares future unnecessary extra processing for the same word occurrence.

As for the rest of the unknown words, a heuristic was developed applying the known language patterns regarding sound shifting in Viennese, which are reflected as changes in vowels observing the data as textual data type (see Breuer 2021). The heuristic aims at regaining unknown words using the character level approach as part of the preprocessing step. The information is saved as a dictionary data type supported by Python, where the key stands for the vowel in the source language and the value(s) for the possible occurrences in the target language. The method alternates the vowel and checks with the help of the lookup mechanism, if the word appears in the corpus, either in the source or in the target language. In case of a match, the word is added accordingly, and the workflow proceeds to the main translation task via the N-grams.

Special handling is done for unknown words that hold the letter <g>, based on the grammatical structure of verbs in the past participle tense. In Viennese, the form of a verb in this tense tends to be shortened by omitting the following letter <e> (e.g. *gestürzt*, 'fallen', being shortened to *gstürzt* in Viennese). There are two forms where <g> appears; either at the beginning of the sentence, in this case, the whole word holds only one part, or it appears in the middle of the word, belonging to the so-called separable verbs (e.g. *abgestürzt*, 'crashed'). These verbs are created by two parts, where – in certain grammatical contexts – each can be used independently. Each of the parts can occur both in dialect and in its standard German version. After identifying which form the OOV word has, a lookup is conducted for each part of the word. The mechanism is then similar to the previous one and the omitted <e> letter is appended after the <g>, and the newly created word that is strongly believed to belong to the target language is added to the corpus.

This heuristic might not cover all unknown words in an input sentence, however, it greatly contributes to the translator. It establishes a further step towards the computational understanding of a not-standardized language, while this kind of inferring and vowel alternation is done in most cases naturally by German speakers in the Bavarian language space.

4.3.2 Experimenting with German BERT using masks for predictions

For the task of finding possible candidates for the unknown words, an experiment with BERT's capability of predictions was conducted. BERT, which stands for Bidirectional Encoder Representations from Transformers, was developed and published in 2018 using unsupervised deep learning techniques and enables a wide variety of NLP tasks. Deep learning is applied in BERT via artificial neural networks that contain multi-layer transformers. A BERT model is constructed with two steps; pre-training and fine-tuning.

One of BERT's unique training approaches is the Masked Language Modeling (MLM), which can be described as a fill-in-the-blank task. In this case, a model bases its prediction regarding the next suitable word by observing the context words surrounding the mask token (see Devlin et al 2018).

We experimented with BERT's language task supported by the pre-trained models with the 'fill-mask' pipeline.[5] The 'dbmbz/bert-base-german-cased' model was used as the model and as the tokenizer for the pipeline. It provides a further look into the integration of the existing Standard German

[5] https://huggingface.co/dbmdz/bert-base-german-cased [last access 02. 09. 2021]

applications with the particular data of Viennese. The idea behind using it for predicting unknown words is to benefit from the similarity of both languages, especially when the source language lacks resources. The results can lead to a broader overview of the success of such integration and reflect some language phenomena of Viennese in comparison to Standard German.

Only the sentences that contain one or more unknown words are considered for this extra feature. The filled-in mask requires a sentence that contains a mask token [MASK]. The current fill-mask for the pre-trained models supports a mask prediction with a limitation of one masked word per input sentence. Therefore, the language task for such sentences is executed sequentially; for each unknown word, one mask is placed and the rest of the positions are filled with the original unknown words. At the end, the final sentence is constructed by inserting at each position the chosen candidate per mask.

The return value from the pipeline includes a few candidates (normally 3–5) that are model-dependent and sorted by their probabilities. Hence, choosing the best candidate based on its similarity to the original OOV word makes more sense and can improve results, as the decision based on the Viennese model creates a stronger correlation rather than the external Standard German one that lacks the sensitivity for this particular data. The Levenshtein Distance is ideal for implementing such a selection mechanism that is edit distance-based.

4.4 Results of the Translation

4.4.1 Methods for Evaluation

One- dimensional quantitative analysis is not sufficient to assess the properties of the machine translation, due to the particularity of the data and the heuristic method. Therefore, different testing methods were used in order to facilitate a wider understanding of the quality of this machine translation. This includes quantitative as well as qualitative methods, that are especially essential for determining the quality of the heuristic for the out-of-vocabulary.

For general statistics, we refer to absolute translation correctness as a hit/miss rate, i.e., the output sentence is identical to the expected translation, and correct partial translation, which can be measured with the WER score (word error rate) and with BLEU (bilingual evaluation understudy). For this work, the main method for evaluating the quality of translation is chosen to be WER score. The quantitative results refer to case-sensitive as well as case-insensitive. The case-sensitive check aims to measure how well the machine observes the orthographic grammar rule regarding capitalization. This property can be treated as another quality check of the machine. It is, however, not a vital criterion because an external grammar checker is able to perform this task.

The formula for WER calculation goes as follows: $WER = (S + I + D) / N$

where:[6]

S… stands for substitutions (replacing a word)

I… stands for insertions (inserting a word)

D… stands for deletions (omitting a word)

N… the total number of words appearing in the sentence

[6] See https://deepgram.com/blog/what-is-word-error-rate/ [last access 03. 09. 2021]

4.4.2 Test Set

The training set for creating the corpus consists of 100,000 parallel input sentences.

The test set consists of 24,000 parallel sentences, creating a relation of about 80%–20% training/test data (see table 2).

Table 2 *BLEU Score*

WER	BLEU	Case-sensitive	Case-insensitive
4.4731%	≈1	5,096/23,207 sentences incorrect 78.041% correct	3,423/23,207 sentences incorrect 85.25% correct

Such a high average BLEU score indicates that there is a high level of uni-/bi-/tri- and 4-grams correlation and that the data for training and testing might be overfitting. No external model for references is used, but can be included in future work.

4.4.3. Evaluation of the OOV Heuristic

Reviewing the results manually (see table 3), the words that the machine managed to regain via the heuristic illustrate the property of flexibility of this machine translation. It enables the understanding and deeper processing of Viennese, based on the language's patterns and nuances. The majority of the unknown words belong to the Standard German domain, and they are not recognized by the system because they have not appeared in the training set.

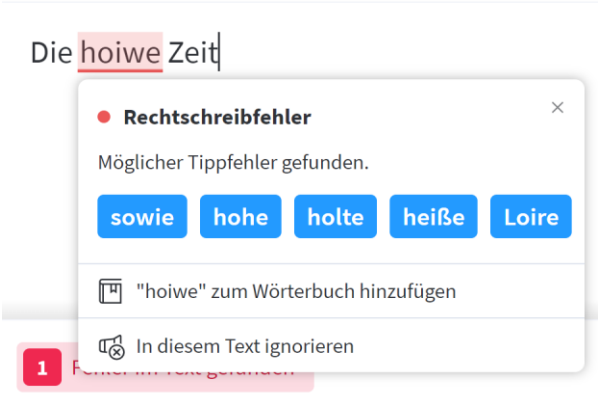
Table 3 *OOV heuristics*

Overall unknown words	Gained back words	Regain rate
4,106	221	5.38%

Some relevant examples: *aamol* (Viennese 'once') was changed to *amol*. This word is an example of adapting to the source language model, from this point it can be further processed by the translation task which will produce *einmal* (Standard German 'once'). It shows the flexibility of the system to handle anomalous inputs. Another similar example shows the capability of the system to cope with different varieties: *hoiwe* (Viennese 'half') was changed to *hoibe* that will be eventually translated to *halbe* (Standard German 'half').

The grammar correction task for *hoiwe* was tested with LanguageTool (2021), as shown in figure 2:

Figure 2 *LanguageTool translation for hoiwe*



This example leads us to the conclusion that external German language models might not be sensitive and adequate enough for Viennese.

4.4.4. Evaluation of BERT's Mask Prediction

The average WER value for the sentences rebuilt with BERT's masks stands at ≈20.63%, whereas for the same sentences without BERT's intervention (meaning the output translation that used only the OOV heuristic) the WER is 14.87%.

Overall only 21 words out of 3,237 processed sentences with unknown words were found to belong to the Standard German domain. This result is very poor, as not even 1% was recognized well for this task.

Overall 211 out of 3,237 sentences scored better than the output translation, i.e., a lower word error rate, which is about 6.5%.

4.5 Discussion of Further Translation Work

The high rate of the quantitative methods confirms the first assumption that an N-gram model with no neural network involved would also return relatively good results. Additionally, the machine managed to identify the need for capitalization for the formal second person singular pronoun *Sie* and *Ihnen* (which are formally identical with the second person plural pronouns, except for capitalization) in most cases.

The quality of the results of the correct partial translation has reached a point from which external models and libraries can be integrated to enhance the final output translation, e.g. autocomplete correction, grammar correction, etc. This also leads to a possible approach for future work, where the output translation of this machine is treated as a pivot language between the dialect and Standard German.

The OOV heuristic has shown the flexibility of the system to deal with deviations from the existing corpus by alternating specific vowels based on known patterns and nuances of Viennese. Also if the rate of regaining unknown words is pretty low (as described about 5%), it provides the system with the ability to handle language varieties in a way that is still not fully supported with the existing models and transformers.

From the results, it can be concluded that almost none of the candidates that were offered by BERT's prediction mechanism was correct, based on the comparison of the WER values.

This feature could be still used as fine-tuning for the words that do belong to the Standard German domain that suffer from misspellings, as the examples above showed.

However, it is clear from the poor results that Viennese poses a language that might be very similar to Standard German and have multiple intersection sets with it, yet consists of very particular figures of speech, idioms and ways of expressions. These do not necessarily intersect with Standard German. Most of the examples where no match is found between the best candidate offered by BERT and the intended word demonstrate this particularity. Thus, this implies that the current NLP tools and models for Standard German might not be fully adequate for handling dialects in general and Viennese in particular. While the equation of the Viennese dialect with Standard German might be problematic, the grammatic, syntactic and morphologic similarities between these two varieties can and should be utilized. This leads to a possible future need for more specialized or extended models that include trained data based on dialects.

5. Conclusion

This paper aimed to describe some of the core functionalities and innovations of Transcribe. Starting with its methods for audio processing and the visualization of audio data, continuing with Transcribe's utility for linguistic transcription, especially in the context of larger, collaborative research projects with different foci, and concluding with a possible application for machine translation. We are confident that further possible applications can and will be combined with Transcribe. Additionally, we want to expand Transcribe by several other functions, among them the possibility to easily connect it with a cloud-based back-end. We hope that this will lower the threshold of using it in the context of smaller-scale, decentralized research groups, who might not be able to afford hosting a server, but still want to work collaboratively on transcription. Additional features are still being conceptualized, developed and tested, but suggestions and comments from the linguistic and IT-community are welcome. The aim of Transcribe is not just to be a tool to enable researchers and any others working with language data to transcribe their data as convenient and possible and as detailed as necessary, but also to highlight some of the methodological implications of the act of transcription.

Transcribe is at this point still a work in progress, with a first stand-alone version to be released soon. The software is open-source and free, and can the code can be found in the GitHub Repository of the SFB DiOE (see DIOE 2020). Additionally, the public release of Transcribe, which is planned for 2021, will be announced on the website of the SFB DIOE.

References

Audiocogs 2015: *Ogg.js. GitHub Repository*. https://github.com/audiocogs/ogg.js [last access 04. 02. 2020].

Audioplastic 2017: *FFT Benchmarks. GitHub Repository*. https://github.com/audioplastic/fft-js-benchmark [last access 04. 02. 2020].

Breuer, Ludwig Maximilian 2021: *„Wienerisch" vertikal. Theorie und Methoden zur stadt-sprachlichen syntaktischen Variation am Beispiel einer empirischen Untersuchung in Wien*. Dissertation at the University of Vienna

Brook, Corban 2017: *DSP.js. Digital Signal Processing for Javascript*. GitHub-Repository. https://github.com/corbanbrook/dsp.js/ [last access 14. 02. 2020].

Chammas, Adriana, Quaresma, Manuela and Mont'Alvão, Cláudia 2015: A Closer Look on the User Centred Design. *Procedia Manufacturing* 2015/3: 5397–5404.

Debasmita, Saha, Ardhendu, Mandal, and Pal, S. 2015: User Interface Design Issues for Easy and Efficient Human Computer Interaction: An Explanatory Approach. *International Journal of Computer Sciences and Engineering* 2015/3: 127–135.

Demorest, Paul 2007: GPU Benchmarking. https://www.cv.nrao.edu [last access 07. 02. 2020].

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton and Toutanova, Kristina 2018: Bert: *Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805.

DiOE 2018: Overview. https://dioe.at/en/article-details/ [last access: 03.09.2021]

DiOE 2020: Github Repositories of the SFB: Deutsch in Österreich. https://github.com/german-in-austria [last access 03. 09. 2021].

European Commission 2016: *Open innovation, Open Science, open to the world*. A vision for Europe. Brussels: European Commission, Directorate-General for Research and Innovation. https://dx.doi.org/10.2777/061652 [last access 17. 5. 2018].

FFmpeg 2016: ffmpeg tool (Version be1d324) [Software]. https:// ffmpeg.org/ [last access 12. 02. 2020].

Google 2014: Material Design. https://material.io [last access 01. 02. 2020].

Google 2019: Tensorflow.JS. URL: https://js.tensorflow.org/api/0.13.3/#spectral.fft [last access 07. 02. 2020].

Gossman, John 2006: Advantages and Disadvantages of M-V-VM. https://docs.microsoft.com/en-gb/archive/blogs/johngossman/advantages-and-disadvantages-of-m-vvm [last access 01. 02. 2020].

Heideman, Michael, Johnson, Don and Burrus, Charles 1985: Gauss and the history of the fast Fourier transform. *Archive for History of Exact Sciences* 34: 265-277.

Indutny, Fødor 2017: *FFT.js*. GitHub Repository. https://github.com/indutny/fft.js [last access 03. 02. 2020].

Justice, David 2014: *DrawWave*. GitHub Repository. https://github.com/meandavejustice/draw-wave [last access 20. 02. 2020].

Kendall, Tyler 2008: On the History and Future of Sociolinguistic Data. *Language and Linguistics Compass* 2/2: 332–351.

Khronos Group 2020: WebGL 2.0. *Specification*. https://www.khronos.org/registry/ webgl/specs/latest/2.0/ [last access 22. 02. 2020].

Koehn, Philipp 2009: *Statistical Machine Translation*. Cambridge: Cambridge University Press

Krasner, Glenn E. and Pope, Stephen T. 1988: A Cookbook for Using the Model-View-Controller User Interface Paradigm. *Journal of Object-Oriented Programming* 88/4: 26–49.

LanguageTool 2021: https://languagetool.org/de/ [last access 03. 09. 2021].

Leider, John 2020: *Vuetify*. Material Design Component Framework. https://vuetifyjs.com [last access 01. 02. 2020].

Lenz, Alexandra N. 2018: The Special Research Programme: German in Austria: Variation – Contact – Perception. *Sociolinguistica* 32/1: 269–277.

Linklater, Greg, Marais, Craig, and Herbert, Alan 2018: Offline-First Design for Fault Tolerant Applications. SATNAC 2018, South Africa, 260–265.

Mozilla Developer Network 2021: Range Requests. https://developer.mozilla.org/en-US/docs/Web/HTTP/Range_requests [last access 03. 09. 2021].

Nagy, Naomi and Sharma, Devyani 2013: Transcription. In: Podesva, Robert J. and Sharma, Devyani (eds.): *Research Methods in Linguistics*. Cambridge: Cambridge University Press, 235–256.

National Instruments 2019: Schnelle Fourier Transformation. https://www.ni.com/de-at/innovations/white-papers/06/understanding-fftsand-windowing.html#section--241931811 [last access: 02. 02. 2020].

Nielsen, Jakob and Molich R. 1990: Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. Seattle, 249–256.

Nielsen, Jakob 1999: Do Interface Standards Stifle Design Creativity? Jakob Nielsen's Alertbox, August 22, 1999. http://www.useit.com/alertbox/990822.htm [last acess 12.2.2020].

Nielsen, Jakob 2012: User Satisfaction vs. Performance Metrics. https://www.nngroup.com/articles/satisfaction-vs-performance-metrics/ [last access 17. 02. 2020].

Pfeiffer, Silvia 2003: *The Ogg Encapsulation Format*. Request For Comments 3533. *https://tools.ietf.org/html/rfc3533* [last access 01. 02. 2020].

Raskin, Jeff 1994: *Intuitive equals Familiar*. Communications of the ACM 37/9, 17 https://www.asktog.com/papers/raskinintuit.html [last access 18. 2. 2020].

Rosenberg, Duane 2018: GPU parallelization of a hybrid pseudospectral fluid turbulence framework using CUDA. Atmosphere Journal of Physics 11/2, 178-200. https:// arxiv.org/pdf/1808.01309.pdf [last access: 01 02. 2020].

Sasiki, Kay 2020: Fast Fourier Transform in TensorFlow.js WebGL backend. https:// www.lewuathe.com/webgl-implementation-of-fast-fourier-transform.html [last access 01. 02. 2020]

Trost, Harald 2016: *A Hybrid Approach to Statistical Machine Translation Between Standard and Dialectal Varieties. Human Language Technology*. Challenges for Computer Science and Linguistics: 6th Language and Technology Conference, LTC 2013, Poznań, Poland, December 7-9, 2013. Revised Selected Papers. Vol. 9561.

World Wide Web Consortium 2011: Web Audio. https://www.w3.org/TR/2011/WD-webaudio-20111215/ [last access 03. 02. 2020].

You, Evan 2020: Vue.Js. The Progressive JavaScript Framework. https://vuejs.org [last access 01. 02. 2020].

DOI: https://doi.org/10.14232/wpcl.2022.7.2

Generating preview word maps in the DMW project

Kai-Uwe Carstensen

1. Introduction

The DMW project – funded by the Academy of Sciences and Arts of Northrhine-Westfalia, Germany, and involving the universities of Bonn, Münster, Paderborn and Siegen – is a long-term project (2016-2032) that aims at digitally collecting, analyzing, storing, presenting and preserving dialect data in the western part of Germany (mainly Northrhine-Westfalia).[7] To make this possible, we digitally record spoken data in ca. 800 different places, with four informants per place, each two of them aged '30-45' and '70+', respectively. We use a questionnaire of more than 600 succinct tasks, according to which the informants mostly have to answer open and yes/no-questions, to describe pictures or video scenes with a term, or to read sentences in their non-standard manner of speaking. These data are processed (cutting/segmentation and analysis of the sound files), and presented on dynamically generated maps (so-called *preview maps*) of a digital dialect atlas of the region.

We distinguish between two versions of the digital, dynamic DMW atlas according to the user types addressed: first, a standard, public version (aka "Speaking DMW"), which allows to view the distribution of dialectal variants in space, and to listen to the corresponding recordings (these are either the answers to some question on a "word map", or the Wenker sentences read out by the informants on "Wenker sentence maps"); second, an extended expert version with restricted Shibboleth access, which will also portray the distribution of ca. 1200 dialect phenomena variants (similar to other dialect atlases). Both versions are based on visually presenting the variants in categorized form ("taxates" in Goebl 2010's terminology), yet the types/taxates are determined *automatically* in the former according to the scheme described below, *intellectually/manually* in the latter. In the final phase of the project, some of the phenomena-related data will be evaluated and published as classic, annotated dialect maps of the region.

Preview maps[8] systematically depart from classic dialect maps in at least the following respects. Preview maps are inherently digital and dynamic, i.e., they are instantaneous projections of actual analyzed data generated automatically on-line as a result of a user query. By offering various kinds of selection and presentation options, they are interactive and can be used for visual exploration of *all available* dialect data even by lay people from early on. Yet they lack classic, evaluation-based dialectal annotations ((core) dialect areas, isoglosses etc.) or fully theory-driven data clusters and variant types. In contrast to that, *classic dialect maps* are static (print-oriented), non-interactive, non-explorative maps constructed manually by experts for experts. They are based on time-consuming *evaluation* of – mostly selected – data and portray them as filtered by intellectual considerations and with corresponding

[7] See Spiekermann et al. (2016ff) for general information about the project; for an overview of the workflow, see Carstensen et al. (2020).

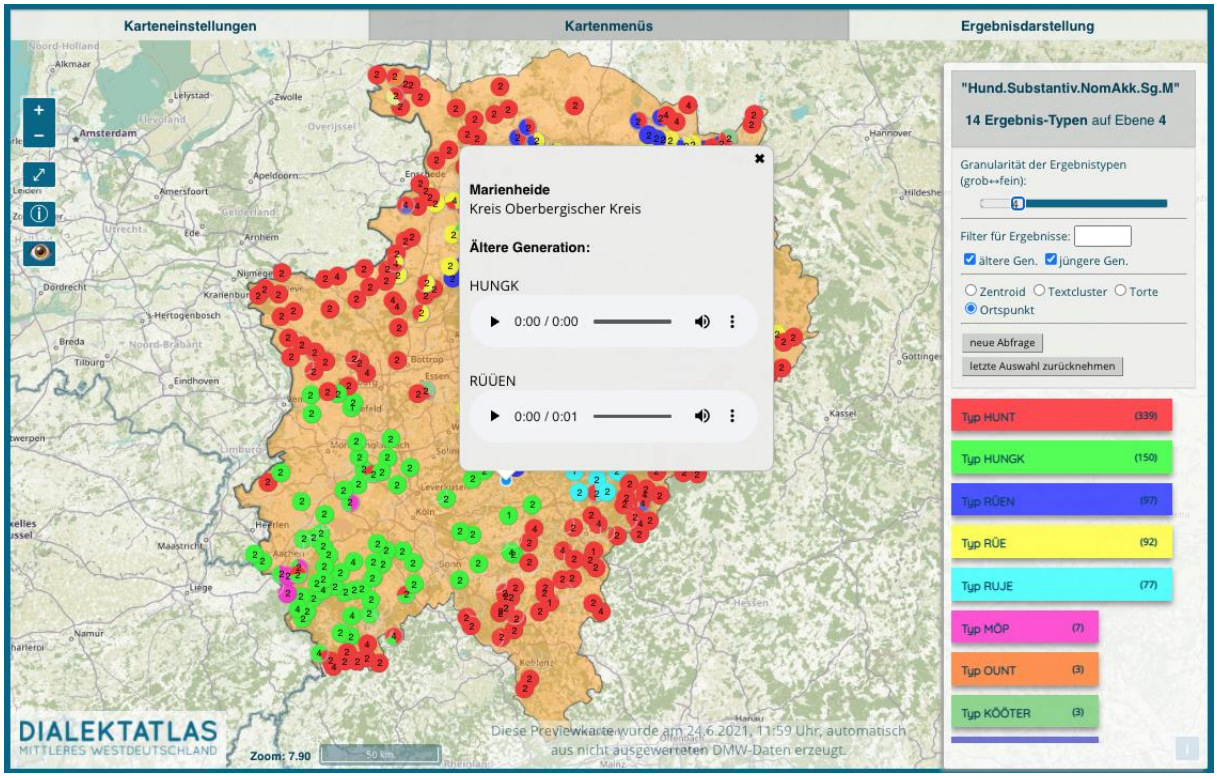
[8] The preview word maps as presented here have been introduced in the SiSAL project (Solau-Riebel – Vogel 2013-2016), albeit in a much smaller scale, and with much more manual data preparation.

Nicole Palliwoda (ed.): *Data Processing and Visualization in Variational Linguistics/Dialectology*. Working Papers in Corpus Linguistics and Digital Technologies: Analyses and methodology. Vol.7. 2022.

dialectological annotations. Apart from different possibilities and habits of information presentation in modern times (non-print, user-friendly, computer-generated, interactive, explorative, conforming to GUI principles), an important disadvantage is the expectable time lag for classic dialect maps, because they require complete data sets for the selected phenomena. This makes preview maps the primary (and for the most part, only) option for the DMW, given the size of the project (number of informants, questions, phenomena) and the maxim to use digital technology throughout.

Irrespective of the classic/non-classic contrast regarding intellectual evaluation, current preview maps are different from most, if not all, computational approaches to dialectology in dialectometrics or geolinguistics (see Lameli et al. 2010 for an overview) in that they do *not* show statistical analyses of more than one item or feature ("global" analyses in Goebl's terms). Instead, they are intended to easily identify aspects of variant distribution in space for *particular* analyses/items, given massive data variation. The present paper describes generating *preview word maps* based on the transcriptions of uttered words like the one shown in Figure 1, to be distinguished from *preview phenomenon maps* showing the variants derived from theory-based analyses (of some phonetic, morphological, lexical, or syntactic aspect of uttered words) that will be developed later.

Figure 1 *Preview map of* 'Hund'



At the beginning, we faced a number of challenges to be met for a success of the DMW project: efficient, error-less, off-line digital fieldwork/interviews/recordings ("exploration"); efficient, error-avoiding, computer-assisted high-quality transcription; effective automatic visualization of dialect data using up-to-date web technology. To exemplify this with current numbers: as of June, 1st, 2022, we have cut and transcribed ~590.000 spoken words of more than 550 explored places, and in the presentation of a word map, the number of variants of a single word can easily reach 100, or even 200 (of already generalized IPA transcripts). Figure 1 shows how our preview maps can give a rough

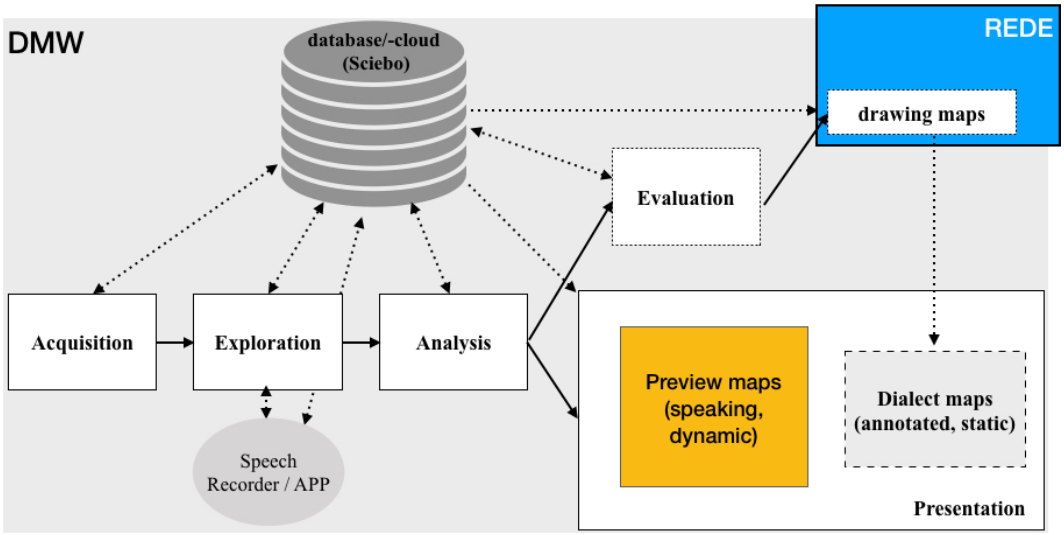
indication of data distribution in spite of massive data variation (in this case, more than 60 already generalized variant types for the word *Hund* ('dog'), reduced to a display of 14 types) by choosing certain clustering and visualization options.

In following, I will first describe the work and data flow in the DMW project, with a focus on the computer-assisted transcription (aspects of the computer-assisted exploration are further discussed in Gehrke et al. 2020), introducing our literal "popular" transcription (POP). After that, I will give a detailed account of our POP-based algorithm for handling massive dialect data to achieve effective visualization. Finally, I will elaborate on the technical aspects of preview word map generation in the DMW project for the standard version, also showing the possibilities of its interactive and explorative use.

2. Work and data flow in the DMW project

Basically, technical aspects in the DMW project can be described as entering, retrieving and modifying data of our central database (apart from storage of audio data in the scientific cloud Sciebo). From early on, handling DMW data was conceived as human-computer interaction via (graphical user) interfaces: the *acquisition interface* for handling contact (action) information for places to be explored; the *exploration interface* for uploading personal information and (meta-)data of the interviews; the *analysis interface* as a work area to handle informant data (cutting sound files, transcribing utterances, and handling phenomena aspects); the *map (presentation) interface* presenting preview maps for selected queries of some user. Figure 2 gives an overview of the simplified work and data flow structure of the DMW project.

Figure 2 *Global view of work and data flow in the DMW project*



Note that production/drawing of classic, static dialect maps is planned but not yet done, hence the different graphical appearance as dashed components. It requires both a stage of intellectual evaluation of the analyzed data, and manual production of maps (which will be done with the map drawing tools provided by REDE, see Schmidt et al. 2008ff).

Over the years, additional functionalities and tools requiring database interactions were developed. One of them is the *IPA repair tool* that allows to easily spot transcription errors by listing selected data,

and to quickly correct them via the offered shortcuts to the corresponding (parts of) the analysis interface (see below for more information). While the interfaces were originally viewed as wholistic web applications covering the central aspects of our work flow, we therefore rather built a structured internal web presentation for our project (with corresponding work flow areas) in which the interfaces, but also other web-based functionalities, were placed accordingly.

2.1 Acquisition

In general, we require exploration places to be a subset of the classic Wenker places constrained by a certain range of inhabitants (500 to 8000) and some scheme of distribution: we use a raster overlay of our exploration area to arrive at an equally distributed selection of places to be explored, each partner university being responsible for a certain part. The acquisition interface involves a WebGIS application showing area, raster and Wenker places, as well as filterable layered information about the places (aspects of the contact status and (partial) exploration status of age groups to be explored). A click on a place accesses the acquisition interface proper which allows to enter and view contact data as well as time stamped acquisition actions (contacting persons via phone or mail, placing advertisements, distribution of informant questionnaires etc.). Once established, this interface provided a remarkable facilitation of acquisition planning and coordination.

2.2 Exploration

For the digital explorations, we use the SpeechRecorder® app (Draxler and Jänsch 2004, 2019) with which the answers of the informants to the questions of the exploration questionnaire are automatically cut and systematically stored – an enormous saving of time, and hence, man power, for the subsequent analysis stage. Unfortunately, the SpeechRecorder was designed for use as a laboratory software, which does not guarantee uniqueness of identifiers or easy adjustments of the built-in questionnaire when used distributedly in different locations, on various computers, and in the field. We needed to come up with a quick solution for these problems since explorations were supposed to start immediately.

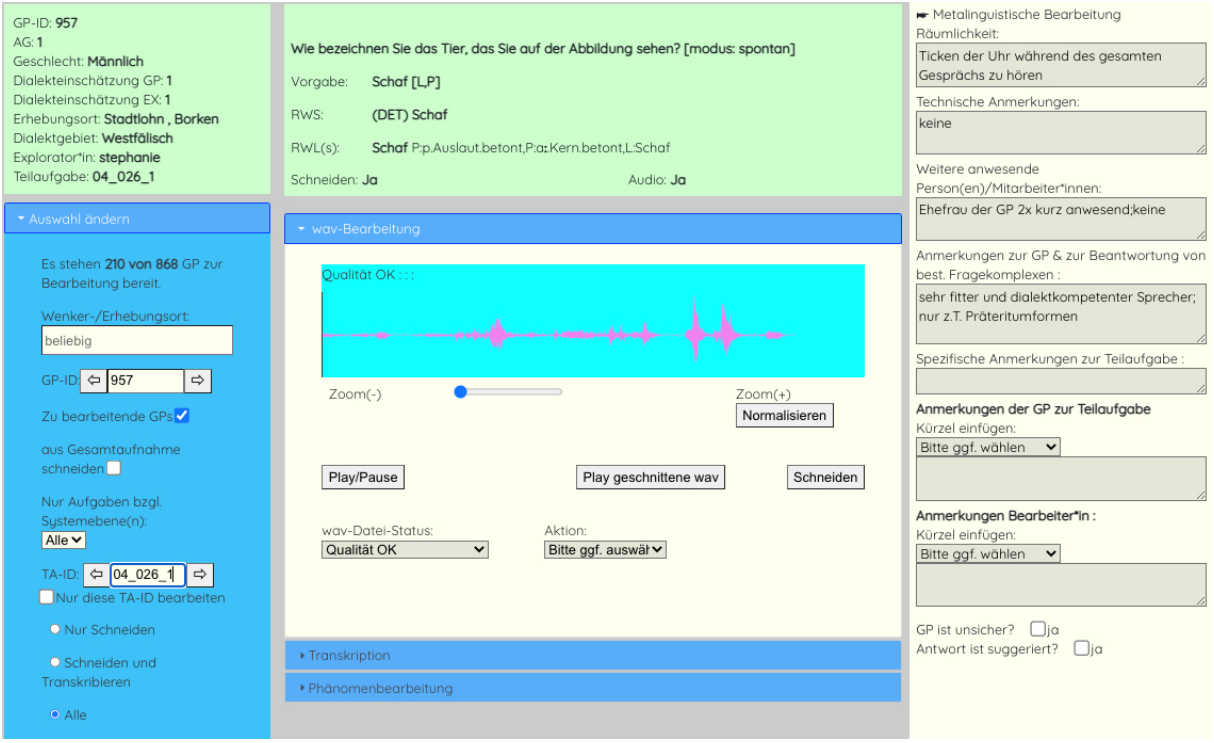
We opted for dedicated semi-automatic work flows using python scripts to handle the necessary organizational data structures. The first python script transforms the original exploration questionnaire into the XML format required by the SpeechRecorder. It is applied semi-automatically every time the questionnaire is modified (which happened quite often in the beginning of the project). Then, before an exploration, each explorator of university with id U requests a “SpeechRecorder project” for an informant with –a unique– identifier ID. This automatically fires the second python script, creating unique project files (using U and ID) and bundling them into a downloadable zipped SpeechRecorder project importable by the SpeechRecorder. U and ID, together with the question numbers of the “SpeechRecorder Script”, are used to name the corresponding audio files of the answers. After the exploration, the folder containing audio files is uploaded into the cloud, accessible for use in the analysis and map interfaces.

The exploration interface is used by the explorator to store information noted in the *exploration protocol* (remarks about the setting –ambience, technicalities, other persons present, characteristics of the informant– and about aspects of specific answers). Even before the exploration, it is also used to enter all information about the informant (a prerequisite for the exploration).

2.3 Analysis

Analysis in the DMW project roughly divides into treating sound aspects of an answer, transcribing the relevant answer word(s), and handling phenomena information. The analysis interface (see Figure 3) allows to make certain selections, e.g., of place, informant, task, analysis part to deal with (in the blue area) or to view relevant (meta) information, for example, about informant, exploration setting, or task/analysis like question asked, and words to be analyzed (in the green area).[9] While wav-handling and transcription (see below) are indispensable for the generation of preview maps, phenomena handling will be performed in later stages of the project. For aspects of metalinguistic information (handling) see Gehrke et al. (2020).

Figure 3 *Analysis interface*



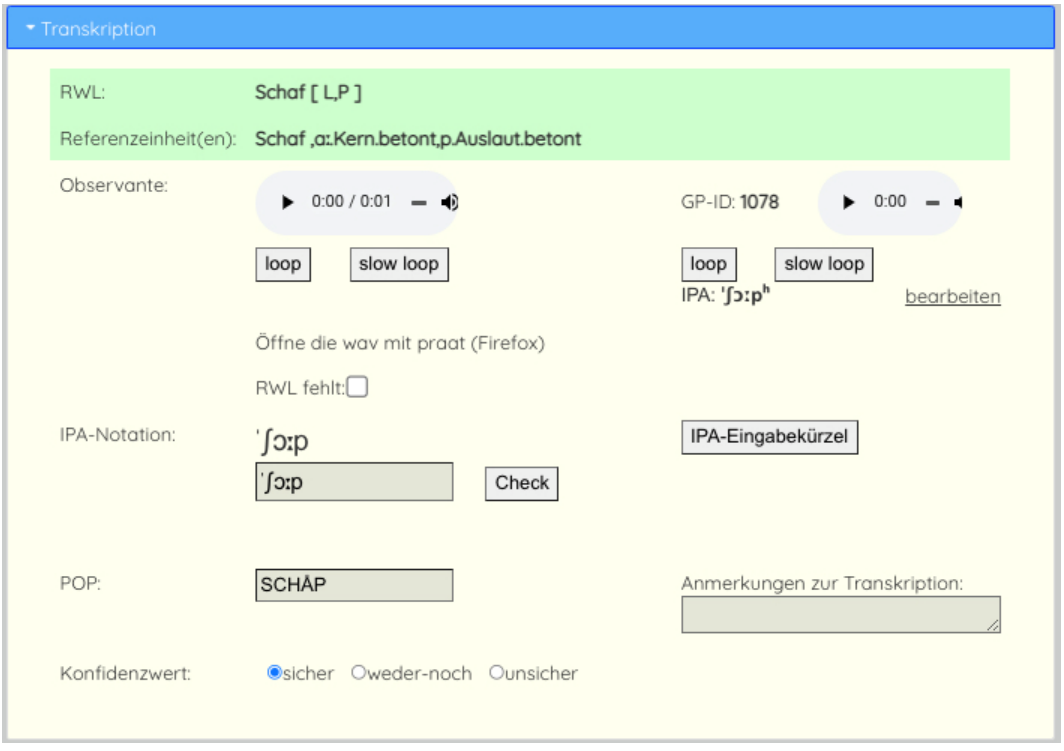
Sound-related analysis (in section “wav-Bearbeitung” of Figure 3) involves cutting audio files for audio presentations on preview maps (for users to be able to hear the non-standard pronunciation of a word, uttered as part of an answer to some question of the questionnaire). We specify the range of words to cut as so-called RWS (**R**elevant **W**ord(s) for cutting (=**S**chneiden)) and also present the information about the words to be analyzed as so-called RWL (**R**elevant **W**ord(s) for **L**inguistic analysis). For example, we might elicit a prepositional phrase *in the barn* with a question *Where does the farmer store the hay?*. Although we could be interested in various phenomena on distinct linguistic levels addressing different RWLs (say, lexical and phonetic aspects of RWLs *in* and *barn*, syntactic aspects of *in the* or *the barn*), and might want to present some of them audibly, we decided to just cut *one* audio file (containing the span of words with variants of all RWLs, in this case, *in the barn*) for each (sub)task as RWS, for economy reasons. Apart from that, the RWS of a single-noun RWL may be specified as having to include

[9] Task handling is restricted, however. To prevent certain analysis artifacts (“analyzer isoglosses”), each analyzing person is exclusively assigned a number of tasks by the coordinator.

the determiner of the noun, if uttered (see Figure 3). Sound-related analysis then means to perform some action (cutting the RWS, optionally improving sound quality), and to store relevant information (e.g., about the quality/status of the audio file or the answer). Note that an answer might be difficult to hear or identify (in case of multiple speakers or multiple different answers), or be lacking for different reasons (no or wrong answer given, question not asked due to unfinished interview, or question not in questionnaire at interview time).

The transcription part of the analysis interface cycles through the RWLs of the current task/question presenting the corresponding observed variants ("observants") for phonetic analysis in each case. Transcription is different from simply entering IPA characters (say, via the keyboard) given the audio file, and is rather realized as a computer-assisted process involving a dedicated set of functionalities, with *IPA transcription tool* (ITT) referring to the area of the analysis interface in which this happens (see Figure 4). The ITT allows to loop both through the current observant (even in reduced speed) and the observant of the other informant of that place (both helps to identify the specifics of the pronunciation). Phenomenon-related information is displayed for the analyzing person to know what to listen to in particular. Transcription is always *computed* given the input into the corresponding field (i.e., transcription is semi-automatic), as detailed below. We use a literal "popular" transcription (so-called POP) as a readable version of an observant to be displayed on preview maps (see below). These POPs are computed *automatically* from the computed IPA of the input.

Figure 4 *The IPA transcription tool (ITT) as part of the analysis interface*



The phenomena handling part of the analysis interface will be used to deal with the ~1200 phenomena to be investigated in the DMW project. Phenomena typically address *parts* of RWLs that have to be identified given the RWL transcript (or the RWS, mostly in the case of syntactic phenomena of non-

transcribed observants). After analysis, the variants of some phenomenon's reference entity will be displayable on corresponding maps in the map interface. This part is still under construction, however.

2.4 Presentation

The results of the analysis of RWLs, as well as the RWSes, are presented on dynamic, unevaluated preview maps as "Speaking DMW" (also collectively called "Atlaskarten"/"Atlas maps").[10] We distinguish two versions of the map interface. First, the standard version for the presentation of word maps (visually displaying the variants of words), and of the Wenker sentences (both with a "click-and-hear" facility). Second, the "expert" version (only accessible via restricted, universitary Shibboleth access) showing variants of phenomenon-related RWL parts.[11] By definition, the preview maps involve no further evaluation, and can therefore be generated as soon as there are analyzed data. Correspondingly, the preview maps directly reflect the progress of the project. Only for the later stages of the project, we plan to evaluate (selected) data in order to present classic static annotated dialect maps for the explored region. Such maps will be constructed with the facilities provided by REDE (Schmidt et al. 2008ff).

3. Transcription in the DMW

3.1 Phonetic transcription in the ITT

Input to the ITT can be either letters, that is, characters of the set [aeioubdfghjklmnprstvwxyz](basic IPA symbols). Any other IPA (diacritic) symbol is specified differently, often in more than one way. In general, we only consider a selected set of IPA symbols (but note that we also code syllable boundaries, ambi-syllabic consonants, primary/secondary stress, and syllabic consonants).[12] To code them, certain textual entries can be made to specify IPA symbols ("sch" → [ʃ], "tsch" → [tʃ], "t-sch" → [t͡ʃ]), double characters either specify long phones ("aa" → [aː]) or ambi-syllabic consonants ("tt" → [t̩]), and certain added characters specify slight variants ("e#"- > [ə], "r#" → [ɹ]). As an alternative, SAMPA (see Wells 1997) character(sequence)s can be used ("S" → [ʃ], "E" → [ɛ], "6" → [ɐ]), also in textual combinations ("t-S" → [t͡ʃ]). In the ITT, there is a help button listing all available textual, SAMPA, or combined character sequences coding IPA symbols.

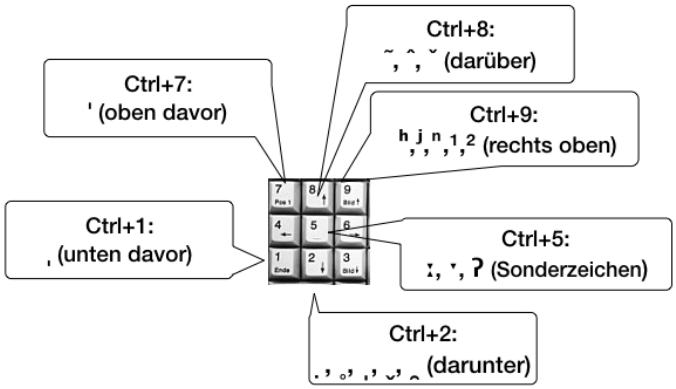
While they can be specified with SAMPA (e.g., "t_h" → [t^h]), there is an additional entry scheme especially for diacritics, based on the position of digits on the numeric keypad (see Figure 5): By Ctrl-entering a digit, the diacritics associated with digit's position relative to "5" are offered for selection (for example, the nasality diacritic is "directly above" a consonant phone, hence the use of 8, directly above 5, to access the corresponding set of diacritics). By Ctrl-entering other characters, phonetic variants of the character are presented for selection (for example, "Ctrl + ö": [øœɜ], "Ctrl + r": [ɹɻʀʁ]). Combined with an auto-completion mechanism that uses the IPA entered so far and the known IPA variants for the RWL (so far, from the database), the ITT therefore offers a quick and non-annoying way of entering IPA symbols for transcription.

[10] This is the external terminology for what is project-internally called "map interface".

[11] In addition to that, simple dialectometric reference point maps (see Goebl 2010) can be generated.

[12] (Non-)syllabicity is coded redundantly and automatically, as are affrication arcs.

Figure 5 *Ctrl-entering diacritics (from German tutorial)*



The *quality* of the transcriptions is secured on many levels, starting with the slow-loop option to listen to the observant, and ending with a global transcription review process whose start is scheduled in 2025. In between are further aspects of quality assurance.

Most importantly, a red “Check” button has to be pressed obligatorily before transcript storage. In that case, a number of tests are performed to check for obvious or possible transcription errors. *Actual errors* (clearly wrong input) lead to corresponding messages, while preventing transcript storage. To detect them, this requires some effort using regular expressions. For example, to discover missing syllable boundaries or stress markers, violations of phonotactic rules in a syllable (based on the sonority hierarchy of the involved phones) have to be uncovered. *Possible (typing) errors* lead to warnings (e.g., if rarely used SAMPA uppercase letters like A, M or V have been entered). Only in situations without actual errors the button turns green, and storing is possible.

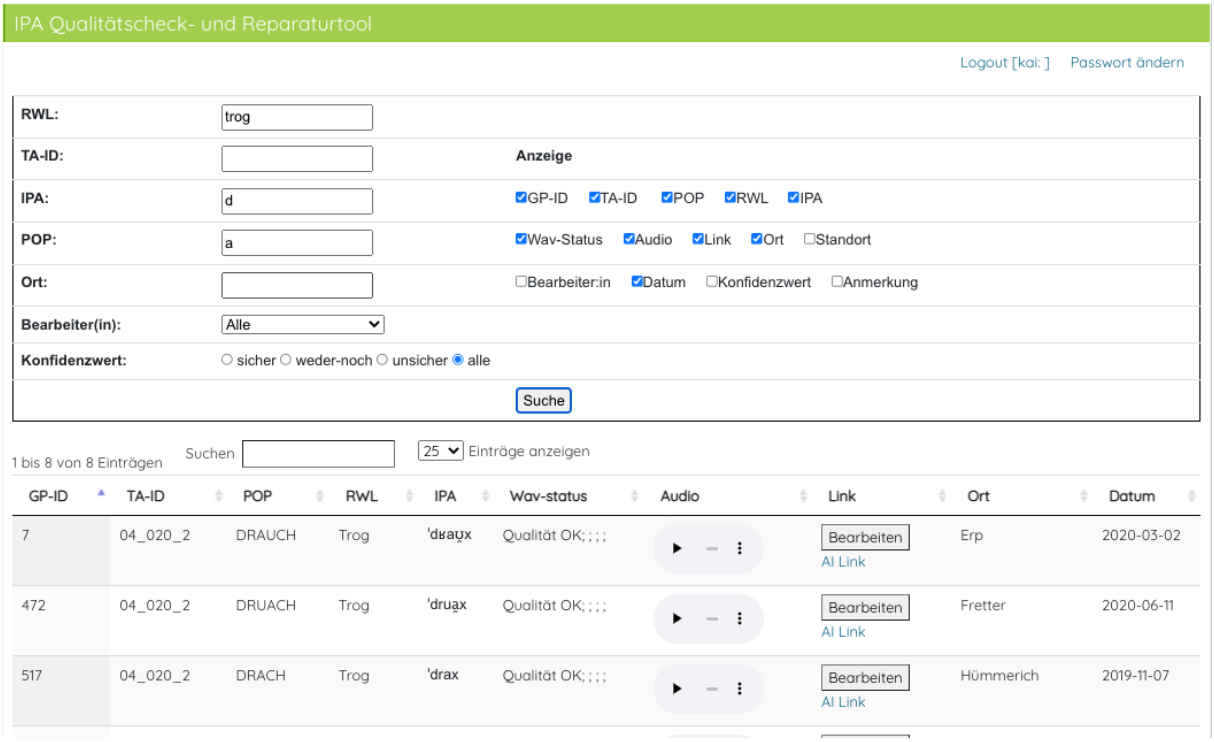
Each time the analyzing person hits Return, performs Ctrl-digit entering, or hits the Check button, the IPA is computed and displayed. At the same time, the literal POP transcription (see below) is computed. Both displays can be used as a feedback about the correctness of the intended transcription.[13]

Finally, the analyzing person has to specify her confidence with a three-valued option (confident, inconfident, in-between). Only then can the transcription be stored.

Outside the ITT, quality can be checked with the preview maps: if there is an outlier in some homogeneous distribution of variants, this is at least a hint to examine the corresponding transcription of the outlier. Likewise, the variants on the preview maps can be filtered by confidence value to identify unconfident transcripts. For both cases, there is an *IPA Repair tool* that offers the possibility to quickly enter an RWL and (at least either) a part of the IPA or POP, or of the place name. As an example, Figure 6 shows the query “list data for ‘Trog’, where IPA contains ‘d’ and POP contains ‘A’” and part of the table of the corresponding results. For easy inspection, the result section is sortable by a click on one of the column titles, and can be filtered by using the full-text search field. The observants can easily be listened to, and links to the corresponding instantiations of the ITT – and also the whole analysis interface– for immediate modification (“Bearbeiten”)/correction are provided.

[13] It should be obvious that transcription in the DMW project requires intensive training. We have Sharepoint pages introducing transcription basics (codes, use of ITT, DMW-specific transcription rules etc.). Above all, there is an IPA transcription *training tool* that mimics the ITT and allows to perform transcription with selected tasks and test data.

Figure 6 *The IPA Repair tool of the DMW project*



3.2 Literal transcription: POPs

Especially due to the many non-expert users of the DMW system, a transliteration of the IPA transcripts of utterances to their readable textual representation is mandatory at least for the word maps. We decided to use the *Hamburgian transcription conventions* ("Hamburger Transkriptionskonventionen", HTC, see Bieberstedt et al. 2016) for this purpose, whose maxim is to "[achieve] an exact reproduction of pronunciation as much as possible, while providing good readability, ensured by using the standard alphabet and a minimum number of special characters (only å)" (Bieberstedt et al. 2016:421, my translation). We call the resulting literal transcripts *POPs* (from "popular"), and write them in uppercase letters.

It is a characteristic of literal transcription that diacritics and prosodic markers are left out, and that certain non-ASCII symbol(sequence)s are systematically mapped on certain character(sequence)s. Examples of the HTC are [ɛ] →<E>, [ɔ] →<O>, [ɔː] →<Å>, [ɔːː] →<ÅÅ>, [ɔi̯] →<EU>, [ʃ] →<SCH>, [d͡ʒ] →<DJ>, [ŋ] →<NG>. Other rules determine how to transcribe syllabic consonants K (as <K'>) or how to mark relevant syllable boundaries (with <->). A large part of the HTC is concerned with the textual marking of vowel length (e.g., when to mark shortness with a double consonant: [kɪn̩ɐ] →<KINNER>, [lʊft] →<LUFT>/*<LUFFT>, [ʃvɪmt] → *<SCHWIMT>/<SCHWIMMT>). On the whole, the HTC provide a good standard for the task of literal transcription, and have been implemented in our system (although automatic realization is less than perfect). It must be noted that the HTC, as our POP conventions, have three main disadvantages, however.

First, they were designed especially for the Hamburgian Low German, while half of Northrhine-Westfalia covers Middle German areas showing regional phonetic phenomena not covered by the

conventions, for example, the so-called “Tonakzent” (phonemically relevant intonation), the voiced velar fricative ([ɣ]), voiced labio-velar approximants ([w]), or different variants of /r/.

Second, their intended use was for directly/manually transcribing spoken dialect, the focus being on the identification of the words uttered. Because of that, they include rules that guarantee recognition of the lexeme, not only of the phonation: for example, while [ts] could be transcribed as <TS>, it has to be transcribed as <Z> at the beginning (as in [t͡saːn] → *Zahn* (tooth)), and as <TZ> at the end of the word/syllable ([kat͡s] → <KATZ>). This standard-language oriented transcription is most obvious in the double-consonant rule following the “morphological principle” in orthography: a consonant is written twice because the lexeme shows this duplication (for example, [brɛnt] is transcribed as <BRENNT>, not as <BRENT>, because the verb is *brennen* (burn)).[14] Similarly, the final unstressed central vowel [ɐ] is to be transcribed as <ER> (but as <A> after [ɛ] or [ee]). Since standard words are the starting points of word maps (“Holz”, “Zahn”, “Feuer”), all this would not have been necessary in the DMW. On the other hand, and thereby different from standard language, the conventions expect an additional hyphen in the case of syllable gaps as in *inadequate*: <IN-ÄDIKWET>. Unlike handling [ts], final [k] is not to be transcribed as <CK> (as in *dick* and *Stock*), but as <KK>. Furthermore, while [ɛ] is mapped to <E> by default, [ɛi] is to be transcribed as <ÄI> (so-called “Hamburger Diphthongierung”). This mixture of rules, adaptions to standard writing, and exceptions, complicate the automatic/ digital use of the conventions, which raises the possibility of errors and might slow down performance.[15]

The third disadvantage concerns the use of <E> and <O> for [ɛ] and [ɔ], respectively. Again, while this may be sufficient for literal transcription (and is also motivated by easy recognition of the words in their standard writing), it might be regarded as unsatisfactory in a dialect atlas due to the elimination of interesting variation.[16] For this reason, the map interface offers an option to select “narrow” literal transcription (“lautnahe Transkription”) in which [ɛ] is mapped to <Ä> (not to <E>), [ɜ]/[œ] to <Œ> (not to <Ö>), and [ɔ] to <Å> (which codes [ɔː] by default; therefore, not to <O>). Further candidates for narrow transcription could be [ʒ] (to be transcribed as <J> according to the HTC, and hence confusable with the transcription of [j]), [ɣ] (not as <CH>), or characteristic variants of /r/.

[14] The adaptation according to the morphological principle is (and can be) done only for the current RWL (and its standard form). Therefore, the variant DACHJESCHOS of *Dachboden* only has one final “s”, although JESCHOS corresponds to *Geschoss* with “ss”. The missing implicit human-in-the-loop, needed for aspects like these, could be regarded as another disadvantage of the HTC for our automatic literal transcription. Note, however, that the authors of the HTC are open to adaptations and modifications of the conventions in principle (Jürgen Ruge, pers. comm., see also Ruge 2019).

[15] In some cases, the HTC are inconsistent (in their goals) or incomplete: non-initial [ts] would lead to <WIRTZHAUS>, <MIITZHAUS>, <HOLTZ> for corresponding variants of *Wirtshaus*, *Mietshaus*, and *Holz*.

[16] Especially because phonation is indicated via POPs in preview maps, and because the IPA transcripts are not shown in the standard (lay)version of the map interface.

4. Handling massive variation in the DMW data for the presentation of preview maps: the DMW POP algorithm (DPA)

4.1 The challenge

For typical words, dozens of variants (IPA- types) can easily be observed (e.g., for the item *Kette* ('chain')) in our data, especially as our region comprises both Low German and High German parts. The situation gets worse if there are lexical variants of a word (for *Mädchen* ('girl'): added variants of *Deern*, *Wicht* and *Lüüt*), in which case the number of IPA types can go into the hundreds. Although literal transcription abstracts from specific pronunciation aspects, the number of POPs for an item/RWL still often is very high.

This presents several problems for a system that aims at generating user-friendly visualizations. Above all, there is the sheer *size* of data variance: Even with a smaller set of POP types, the number of signifiers is still too large for effective visualization of the data. Without further intellectual categorization of the signifiers, it is therefore necessary to add an automatic clustering step to prevent cluttering the preview maps with symbols. Then there is the problem of *symbolization* of signifying, say, more than 3-5 variant types. Existing dialect data visualization systems use specific symbols, color, or a mixture of both for that purpose. Especially when using symbols, they run into the problem of visual confusion, however, especially if the symbols are similar in some respect (e.g. circles containing lines representing different hachures).

The prime challenge therefore was to find a scheme of automatic categorization (a "variant/POP type mechanism") that replaces intellectual classification as effectively as possible. This excludes simple Levenshtein-based mechanisms as they are linguistically neither motivated nor constrained. Development time considerations forbade diving into the field of possible improvements of such methods (see Wieling et al. 2011). Run time considerations ruled out too complex computations, and user interaction requirements called for a non-continuous, qualitative and easily understandable (usually hierarchical) category structure. In general, this excludes quantitative clustering methods that map variation to a graded structure in which categories may not identifiable ("named"), and which has to be actively manipulated by the user for him/her to gradually select the level of detail for a presentation (as described in Haimerl 2005:544). Our *DMW-POP-algorithm (DPA)* is therefore different and uses a qualitative, multi-layered Soundex-inspired algorithm, which is described in the following.

4.2 Soundex

The Soundex algorithm (see, for example, Wilz 2005) was invented to allow for *phonetic search*, i.e., a phonetically motivated procedure that finds matches for a database lookup (especially of names) despite nonidentical search terms (for cases like Möller/Moeller, Schmidt/Schmitt/Schmied). In general, it operates by mapping phonetically similar names onto common index terms so that lookup reduces to matching index terms, at least for a first selection of probable candidates. More specifically, (one variant of) the algorithm consists of the following steps (simplified):

1. Retain the first letter of the word; map vowel-like letters to zero (A,E,I,O,U,Y,H,W → 0);
2. Map consonant(sequence)s on certain digits (B,F,P,V → 1; C,G,J,K,Q,S,X,Z → 2; D,T → 3; l → 4; M,N → 5; R → 6);

3. Reduce adjacent same digits to one digit (this also holds for the digit of the first letter), then delete zeros;
4. Truncate after three digits; for a shorter word, add zeros until the index term has four characters.

Accordingly, “Wikipedia” is mapped to “W213” (as is, for example, “Wakepod”). There are a number of variants of the Soundex algorithm (e.g., *Metaphone* for English, and “Kölner Verfahren/Phonetik” for German, see Wilz 2005), which mostly use a more sophisticated scheme of the mapping and/or allow for more than three digits (even one for the first letter) to code the index term. Note that one can distinguish between “technical” aspects (e.g., the structure of the resulting index term) and “content” aspects (different kinds of phonetic rules used) of the schemes.

4.3 DPA: technical aspects

The DPA departs in several respects from the basic Soundex scheme. First, it starts with approximate phonetic representations ((partial) literal transcripts of IPA) and, according to this, does not map *words* to approximate phonetic representations of words. Correspondingly, there is no need for sophisticated algorithms accounting for the different textual realizations of phones. Instead, the DPA maps approximate phonetic representations to *generalized* approximate phonetic representations. In particular, this means that the input to the indexing scheme, on the one hand, is an *extended* POP: it may still contain some IPA symbols ([ʃʒŋœɣxχ]) to ease processing (e.g., handling fricative symbols instead of <(s)ch>), and for narrow transcription. On the other hand, it is a *partial* POP because the conventions have not yet been fully applied.

Second, unlike Soundex, the DPA does not use a fixed, one-level set of principles for indexing. Instead, it applies the indexing on six levels at the same time, each with a different set of principles resulting in coding different *levels of generalization* of a POP, the levels being ordered from least general (i.e., only containing POPs) to most general. In the Javascript implementation, this results in an array *Types*, with Types[0] being the (final) POP, and Types[5] being the most general index term. When applied to a set of POPs, the POPs therefore are automatically clustered on corresponding *levels of granularity* of the set, from least granular (most general) to most granular (the level of POPs).

The third departure from the Soundex scheme is the use of a secondary sub-element mapping (on levels $1 < i < 5$). If in an index term set one term is mapped to XXX, and another to YYYXXX or XXXYYYY (on the same level), the second is remapped to XXX. This allows to capture the fact that certain words are sometimes uttered as parts of compounds, and that it makes sense to group them (for example, HEAFSLOOF and LOOF, and HEAPSLAUP and LAUP, respectively). Technically, this requires some checks to at least guarantee a high correctness probability of such automatically determined subpart relationship. For example, too short parts (for a level) have to be excluded, as well as accidental parts, so that, on level 4 for *Laub* (‘leaves’), the index terms of LOOF-AFFAL, HEASLOOF, LOOFHOOF, HEAFSLOOF, HERFSLOOF, LOOFTEPPISCH are mapped correctly to the index term of LOOF. Overall, this mapping scheme reduces cluster numbers per level significantly, without pruning the index terms arbitrarily.

Internally, this multi-layered indexing scheme requires considerable computation effort (sorting, remapping, counting, establishing information structures, relating clusters to their sets of POPs on five

levels). This is why computation is restricted (at the moment, to less than 150 types), and the user is asked to set a filter to arrive at a smaller set of types (this might be changed, however).[17]

It is important to note that this granularity mechanism with its automatic determination of clusters, types, and symbolizations is the central, technical characteristic of the DPA, allowing for user-friendly, explorative, selective viewing of variants as the non-manual solution of the above cluttering problem.[18] In the map interface, variants will be presented according to the granularity level chosen, and can even be restricted to a cluster on that level.

4.4 DPA: content aspects

The content aspects of the DPA consist of applying some version of the indexing scheme on each level of granularity. They can be arbitrarily chosen, and are only constrained by the rule that with every level of granularity above POP level, the principles must become more general. In the current implementation, the following steps of increasing generality are used:

1. Abstract from specific vowels, build vowel categories, here: frontal ("vorne"), medial ("zentral"), back ("hinten") vowels by "v/z/h", correspondingly; mark identity (textual length marking) of following vowel in a vowel character sequence by "=" (else mark following vowel by "_");
2. Abstract from gap marking and vowel length (leave out "-" and "=") and condense consonants CC to C.
3. Abstract from vowel quality/category by simply noting "_"; generalize plosive minimal pairs (P/B by "P", T/D by "T", G/K by "K"), and velar/uvular fricatives by "Ç".
4. Abstract from vowel change (leave a single "_" for vowel(sequence)s)
5. Only mark first POP character.[19]

Table 1 lists the different types of codes collected and generated for the word *Ei* ('egg').

[17] Filtering is possible via selecting a cluster on some level (thereby restricting the set of variants to those of that cluster), or by textually filtering the POPs via some regular expression to restrict the current variant set.

[18] While the content aspects are no less important, they represent a tentative proposal, and are open to debate and modification.

[19] Noting the first character of the POP is more specific than expected. This is intended and –although it could easily be adapted to "only mark first generalized character"– it makes very much sense for cases like *Ei*.

Table 1 *Different types of codes for RWL* ‘Ei’

IPA-types	ˈʔɔɪ̯, ˈʔaːɪ̯, ˈʔaɪ̯, ˈʔaːjə, ˈʔɛɪ̯, ˈʔɛɪ̯x, ˈʔæjə, ˈʔɛç, ˈʔɪç, ˈʔɛːjə, ˈʔɛːʏə, ˈʔɔɪ̯ç, ˈʔæɪ̯, ˈʔɔːə̯, ˈʔɔʏ̯ɛ, ˈʔɑɪ̯, ˈʔɛʏ̯ɐ, ˈaɪ̯, ˈʔɔːɪ̯, ˈʔɑːjə, ˈʔaɪ̯.jə, ˈʔɛh, ˈʔɑːɪ̯, ˈʔɒɪ̯, ˈʔɛk, ˈʔaɪ̯ç, ˈʔeːə̯, ˈʔaːə̯, ˈʔɒːɪ̯, ˈʔɛçk, ˈʔɛːj̯ə, ˈʔɛg̯ɐ, ˈʔɪʏ̯ə, ˈʔɒɪ̯ç, ˈʔeɪ̯, ˈʔɛçɐ, ˈʔʊç, ˈʔɛj̯
Level 0 (POPs)	AI, EI, ECH, AAI, EU, ÄI, AAJE, ICH, ÄÄJE, ÅI, EUCH, ECHER, EK, EH, OI, AICH, EGGER, EICH, ÄJJE, ÄÄCHE, ÅE, OCHE, AIJE, EEE, AAE, ECHK, ICHE, OICH, UCH, EJ
Level 1	z_, zÇ, z=_, v_, z=Jz, vÇ, h_, v=Jz, z_Ç, zXXzR, zK, zH, zGGzR, z_X, vJJz, v=Ɣz, hXXz, z_Jz, z==, zÇK, vXXz, h_Ç, hÇ, zJ
Level 2	z_, zÇ, v_, zJz, vÇ, h_, vJz, z_Ç, zXzR, zK, zH, zGzR, z_X, vƔz, hXz, z_Jz, z, zÇK, vXz, h_Ç, hÇ, zJ
Level 3	__, _Ç, _J_, __Ç, _Ç_R, _Ç_, _K, _H, _K_R, __J_, _, _ÇK, _J
Level 4	_, _Ç, _J_, _Ç_R, _Ç_, _K, _H, _K_R, _ÇK, _J
Level 5	A, E, Ä, I, Å, O, U

For comparison, Table 2 presents the data for the longer word *Dachboden* (‘attic’).

Table 2 *Different types of codes for RWL* ‘Dachboden’

IPA-types	ˈbaɬ.kʰən, ˈbɔlə.kən, ˈʃpiʃɐ, ˈʃpaɪ̯.ʃɛ, ˈʃpeːˌçə, ˈbal.g̊ə, ˈbal.kn̩, ˈbal.kn̩, ˈbaɬ.kən, ˈbal.kən, ˈbʏœ̯n, ˈʃpɪʃɐ, ˈzaʊ̯.lɐ, ˈzuˌlə, buːˈa̯ːdn̩, ˈʃpɪˌçə, ˈbal.kə, ˈʃpiːˌkɐ, ˈdak.rɔʊ̯m, ˈdax.ʃʁoɐ̯tʰ, ˈbaˌlə.kən, ˈbal.kʰən, ˈboːdn̩, ˈbɔɾə, ˈsʏl.nɐ, ˈbaɬ.kn̩, ˈdax.boːdn̩, ˈbal.kʰə, ˈʔuˌlɐn, ˈʔuɲɐn, ˈʔuˌlɐɪn, ˈʔuˌlɛn, ˈʃpaɪ̯.çɐ, ˈʔɔg̊əln, ˈʃpæə̯.ʃɐ, ˈzœlɐ, ˈboːdə, ˈʔɔlaɪn, ˈbøːn̩, ˈʃpaɪ̯.ʒɐ, ˈʃpaɪ̯.ʃɐ, ˈdaːkˌbyɐ̯.dn̩, ˈbɐl.kən, ˈzœlɐ.ɪs, ˈʔɔlɐn, ˈdax.ˌboːdn̩, ˈdax.ˌbɔɾən, ˈbɔɾə, ˈhʊl.dɐn, ˈʔʊl.dɐn, ˈløːf, ˈspiːˌçɐ, ˈʃpaɪ̯.ʒɛ, ˈsœl.dɐ, ˈbʏn̩ə, ˈbal.gn̩, ˈbœd̩n̩, ˈbyːə̯l, ˈspiˑʃɐ, ˈʃpi.ʃl̩, ˈʃpiːˌʃɐ, ˈʃpɪʃa, ˈʔɔb̥əˌhaʊ̯s, ˈzœl.dɐ, ˈbal.g̊ən, ˈʃpiːˌçə, ˈbɔl.kn̩, ˈʃpiːˌçɐ, ˈbyːˌnə, ˈhɔlɐn, ˈgʏɐ̯n, ˈbɑl.kn̩, ˈd͡ʒaʊ̯ˌlət, ˈbœn, ˈʃpɔɐ̯ə, ˈʃpɪçɐ, ˈʔʊə̯.bəˌhaʊ̯s, ˈʔuɲəˌdak, ˈbɔl.kən, ˈdax.lɔʊ̯.kən, ˈbyːˌdn̩s, ˌʔuɲɐn.ˈdaɐ̯kə, ˈbyə̯n, ˈbyə̯.dn̩, ˈʔʊn.dɐm daːk, ˈbɔɐ̯n, ˈbalk, ˈdaːˌbɔd̩n̩, ˈʔoːl.dɐn, ˈdak.byə̯.dn̩, ˈboːˌən, ˈbyːə̯.dn̩, ˈhaːˌnəlt, ˈbɔd̩n̩, ˈdak.ˌboːˌdən, ˈbyːə̯.nən, ˈzœla, ˈbuə̯.dn̩, ˈʃpaɪ̯.ʃə, ˈdaɣ.boːm, ˈʔoːl.dɐ, ˈʔoːˌlə, ˈboɐ̯.dn̩, ˈbyːə̯n, ˈzœlɐɾ, ˈdak.ˌkaːˌmɐ, ˈʔɔlɐɪn, ˈbʊʊ̯.dn̩, ˈbaɬə.kə, ˈdak.bɔɲn̩, ˈkʰɛlɐ, ˈdax.ˌstuːˌbə, ˈʃpiːˌʃɐx, ˈʃpaɪ̯.çə, ˈʃpi͡tsˌbɔɲn̩, ˈbʊɾə, ˈʃpɪʃə, ˈʃpɪʃɔɐ̯, ˈdak.bɔd̩n̩, ˈbʊd̩n̩, ˈzɔlɐ, ˈsœlɐ, ˈdaɬaːˌmɐ, ˈbœːn, ˈbœːˌdn̩, ˈbyə̯.nə, ˈdœn.t͡sn̩, ˈzɔl.dɐ, ˌʔʊɲɐm.ˈdaːˌkə, ˈboːˌrə, ˈdiːˌɬn̩, ˈdak.ˌbɔn, ˈboːˌdən, ˈʔoːn.ˌdɔx, ˈʃpɪɐ̯ɐ, ˈspaɪ̯.çə, ˈbʏn.ʃən, ˈdaːkˌboːˌdn̩, ˈʃpeːˌçɐ, ˈʔʊɾə, ˈzœlə, ˈdax.ˌboə̯.dn̩, ˈbal.kˈn̩, ˈdax.ˌbo.dn̩, ˈdaːk.ʁaʊ̯m, ˈkʰaːˌmɐ, ˈdax.jə.ʃɔs, ˈdax.g̊aɱɐ, ˈʃpaɪ̯.çə, ˈʃpɛɪ̯.çə, ˈkaːˌmɐn, ˈdak.ˌboə̯n, ˈdax.ˌbøːv, ˈdaːk.buɔ̯n.dn̩, ˈdaːx.kaɱəɹ, ˈdak.boːˌdn̩, ˈdak.ˌbuɔ̯.dn̩, ˈbaɬ.kˈn̩, ˈʔɔɐ̯.vɪx, ˈʃpaɪ̯.ʑə, ˈʃpɪçɐ, ˈʃpaɪ̯.çn̩, ˈʃpaɪ̯.j̊ə, ˈʃpaɪ̯.çɐ, ˈbuɔ̯.dn̩, ˈbɔl.gən, ˈdak.bʊɔ̯.dn̩, ˈdax.ˌboːˌdˈn̩, ˈʃpɪˌçɐ, ˈʃpɪˌçɐ, ˈdax.ˌbɔd̩ən, ˈhɔɪ̯ˌboːˌdn̩, ˈbal.gə, ˈdax.ˌboə̯, ˈhaːˌnəˌbal.kən, ˈdax.kaːˌmɐ, ˈʃpɪçə, ˈʃpɪʃəʁn, ˈʃpɪçə, ˈbiə̯n, ˈbaɬ.kn̩.ˌʁuːm, ˈʃpiːˌʃəx, ˈʃpiˑʃɐ, ˈdak.ˌboːˌdn̩, ˈʃpɪsɐ, ˈʃpɪt͡sˌboːm, ˈdak.ˌkːa.mɐ, ˈdak.ˌbɔd̩n̩, ˈʃpaɪ̯çɐ, ˈʃpɪˌçə, ˈbaːˌkn̩, ˈzœɬ.dɐ, ˈdak.ˌkʰɔɱɔn, ˈdak.ˌbuɔ̯d̩n̩, ˈʃpiˌçə, ˈbaʁən, ˈdak.ˌbœn, ˈʃpʏçəl, ˈdak.ˌkɔɱɔn, ˈdaːxˌkaɱɐ, ˈgyə̯n, ˈdax.ˌbuɾə, ˈdax.ˌbuʁə, ˈdax.ˌbuə̯, ˈʃpiːˌçɐ, ˈbyœ̯n, ˈda.ˌboːdn̩, ˈlynp͡f, ˈlœɣf, ˈvɛɪm, ˈdax.ˌbɔɾə, ˈdax.ˌboːdən, ˈzœɬɐ, ˈbaɬ.k̊ən, ˈbʊɔ̯dˈn̩, ˈbɔn, ˈʔup.kaɱɐ, ˈdak.boːˌdən, ˈkaːˌmɐ, ˈdax.ˌbal.kən, ˈhaːn.ˌʔoːlt, ˈhaːn.ˌʔolt, ˈbyːn, ˈbaʊ̯.bm̩, ˈzœl.də, ˈdax.ˌbuɔ̯.bm̩, ˈʃpiːˌçə, ˈdax.ˌbɛə̯l.dn̩, ˈʃpaɪ̯.ʒə, ˈzœlɐ,

	ˈʃpaɪ̯.çɐ, ˈzoɪ̯ɐ, ˈdaːk.ˌboːm, ˈhaɪ̯.ˌboːˑdən, ˈdaːk.bøːn, ˈʔʊn̩ɐ ˈdak, ˈdaːx.ˌkaːˑmɐ, ˈlɔf, ˈdax.ˌkaːˑmɐ, ˈman.ˌzaːˑdə, ˈdax.ˌbuɯ̯.dn̩, ˈʃpɛɪ̯.ɕə, ˈdak.ˌkaɱɐ, ˈbʏ᷈n, ˈdax.ˌboːn, ˈgʁɔʊ̯.pə.ˌbal.kən, ˈdakʰ.ˌkaːˑmɐ, ˈdax.ˌbɔd̩n, ˈʃpaɪ̯.ʑɐ, ˈbyːᵾ̯n, ˈdax.ˌbuɯ̯n, ˈʔoːˑdə, ˈʃpaɪ̯.ça, ˈʃtro.ˌbal.kən, ˈdaːˑ.ˌkaːˑmɐ, ˈzœːl.dɐ, ˈzœːˑ.lɐ, ˈtaʊ̯.bn̩.ˌʃlax, ˈʃpaɪ̯ːˑ.ʃɐ, ˈdax.ˌbœn, ˈzʏɪ̯ɐ, ˈhiːˑ.lə, ˈʔuᵾ̯.və.nə, ˈdak.ˌbʊᵾ̯.dn̩, ˈhɔɪ̯ç.ˌboːˑ.dn̩, ˈbɔʊ̯.ə
Level 0 (POPs)	BALKEN, BOLLEKEN, SCHPISCHER, SCHPAISCHE, SCHPEESCHE, BALGE, BALKNG, BALKN, BÜÖN, SAULER, SULLE, BUU-ADN, SCHPISCHE, BALKE, SCHPIIKER, DAKROUM, DACHSCHROAT, BALLEKEN, BOODN, BORRE, SÜLNER, DACHBOODN, ULLAN, UNGAN, ULLERN, ULLEN, SCHPAISCHER, OGGELN, SCHPÄESCHER, SÖLLER, BOODE, OLLARN, BÖÖ-N, SCHPAIJER, DAAKBÜADN, SÖLLER-IS, OLLAN, DACHBORREN, HULDAN, ULDAN, LÖÖF, SPIICHER, SCHPAIJE, SÖLDER, BÜNNE, BALGNG, BÖDDN, BÜÜEL, SPISCHER, SCHPISCHL, SCHPIISCHER, SCHPISCHA, OBBEHAUS, BALGEN, SCHPIICHE, BOLKN, BÜÜNE, HOLLAN, GÜAN, DJAULET, BÖN, SCHPORRE, UEBEHAUS, UNNEDAK, BOLKEN, DACHLOUKEN, BÜÜDNS, UNNANDAKKE, BÜEN, BÜEDN, UNDAM DAAK, BOAN, BALK, DAABODDN, OOLDAN, DAKBÜEDN, BOO-EN, BÜÜEDN, HAANELT, BODDN, DAKBOODEN, BÜÜENEN, SÖLLA, BUEDN, DACHBOOM, OOLDER, OOLE, BOADN, BÜÜEN, DAK-KAAMER, OLLERN, BUODN, BALLEKE, DAKBON´, KELLER, DACHSTUUBE, SCHPIISCHACH, SCHPITZBON´, BURRE, SCHPISCHOA, DAKBODDN, BUDDN, SOLLER, DAKKAAMER, BÖÖN, BÖÖDN, BÜENE, DÖNZN, SOLDER, UNGAMDAAKE, BOORE, DII-LN, DAKBON, BOODEN, OONDOCH, SPAISCHE, BÜNSCHEN, DAAKBOODN, SCHPEESCHER, URRE, SÖLLE, DACHBOADN, DACHBODN, DAAKRAUM, KAAMER, DACHJESCHOS, DACHGAMMER, SCHPAICHE, SCHPEICHE, KAAMAN, DAKBOEN, DACHBÖÖW, DAAKBUONDN, DAACHKAMMER, DAKBOODN, DAKBUODN, OAWICH, SCHPICHER, SCHPAICHN, SCHPAICHER, BOLGEN, DACHBODDEN, HEUBOODN, DACHBOE, HAANEBALKEN, DACHKAAMER, SCHPISCHERN, SCHPICHE, BIEN, BALKN-RUUM, SCHPIISCHECH, SCHPISSER, SCHPITZBOOM, DAK-KKAMER, BAAKN, DAK-KOMMON, DAKBUODDN, BARREN, DAKBÖN, SCHPÜSCHEL, GÜEN, DACHBURRE, DACHBUA, SCHPIICHER, DABOODN, LÜNPF, LÖÜF, WEIM, DACHBORRE, DACHBOODEN, BON, UPKAMMER, DACHBALKEN, HAAN-OOLT, HAAN-OLT, BÜÜN, BAUBM, SÖLDE, DACHBUOBM, SCHPIISCHE, DACHBEALDN, DAAKBOOM, HAIBOODEN, DAAKBÖÖN, UNNA DAK, DAACHKAAMER, LOF, MANSAADE, DACHBUODN, SCHPEISCHE, DAK-KAMMER, DACHBOON, GROUPEBALKEN, DACHBODDN, BÜÜAN, DACHBUON, OODE, SCHPAICHA, SCHTROBALKEN, DAAKAAMER, SÖÖLDER, SÖÖLER, TAUBN-SCHLACH, SCHPAIISCHER, DACHBÖN, SÜLLER, HIILE, UEWENE, DAKBUEDN, HEUCHBOODN, BOU-E
Level 1	BzLKzN, BzLKN, SvLLzR, ΣPz_ΣzR, BzLKŋ, Bv_N, ΣPvΣΣzR, SvLDzR, ΣPz_Σz, DzXBh = DN, SvLLz, Bh = DN, hLLzN, ΣPvΣΣz, BvN, BzLGz, ΣPvXXzR, BzLKz, Bv = _N, ΣPz_Çz, ΣPz_ÇzR, Lv = F, BhDDN, Bh_DN, ShLDzR, DzXBhRRz, DzK-Kz = MzR, ShLLzR, ΣPvÇzR, ΣPvΣz, ΣPz_Jz, ΣPv = ΣzR, DzKBh_DN, ΣPvΣzR, BzLLzKzN, BhRRz, hLLzRN, Bh = Dz, h = LDzN, h = Lz, DzKBhDDN, Bv = N, DzKBh = DN, DzXBhDDzN, ΣPvXXz, DzXBh = N, BhLLzKzN, ΣPz_JzR, ΣPvΣL, Bv = Nz, Gv_N, BzLK, Bv = _DN, DzKBh = DzN, h = LDzR, ΣPv = ΣzX, Bv_Nz, Bh = Rz, DzXBh_DN, Kz = MzR, Dz = XKzMMzR, DzXBh_, DzXKz = MzR, ΣPvÇz, DzK-

	KhMMhN, ΣPv = ÇzR, DzXBzLKzN, Bz_BM, ΣPv = Σz, ΣPz = Jz, ΣPz = Σz, Sz_LzR, ShLLz, Bh = -_DN, ΣPv = KzR, DzKRh_M, DzXΣRh_T, SvLNzR, hŊŊzN, hGGzLN, ΣPv_ΣzR, Bv = -N, Dz = KBv_DN, SvLLzR-_S, DzXBhRRzN, HhLDzN, hLDzN, SPv = ÇzR, BvNNz, BzLGŊ, BvDDN, Bv = _L, SPvΣzR, hBBzHz_S, BzLGzN, ΣPv = Çz, BhLKN, HhLLzN, DJz_LzT, ΣPhRRz, h_BzHz_S, hNNzDzK, BhLKzN, DzXLh_KzN, Bv = DNS, hNNzNDzKKz, Bv_DN, hNDzM Dz = K, Bh_N, Dz = BhDDN, DzKBv_DN, Bh = -_N, Hz = NzLT, Bv = _NzN, DzɣBh = M, BzLLzKz, DzKBhN´, KzLLzR, DzXSTh = Bz, ΣPvTZBhN´, ΣPvΣΣh_, DzKKz = MzR, Bv = DN, DvNZN, hŊŊzMDz = Kz, Dv = -LN, DzKBhN, Bh = DzN, h = NDhX, SPz_Σz, BvNΣzN, Dz = KBh = DN, ΣPz = ΣzR, hRRz, DzXBhDN, Dz = KRz_M, DzXJzΣhS, DzXGzMMzR, Kz = MzN, DzKBh_N, DzXBv = W, Dz = KBh_NDN, h_WvX, ΣPz_ÇN, BhLGzN, Hz_Bh = DN, Hz = NzBzLKzN, ΣPvΣΣzRN, BzLKN-Rh = M, ΣPvSSzR, ΣPvTZBh = M, DzK-KKzMzR, ΣPz_ΣΣzR, Bz = KN, DzKBh_DDN, BzRRzN, DzKBvN, ΣPvΣΣzL, KzMMz, v_MSzL, DzBh = DN, LvNPF, Lv_F, Wz_M, DzXBh = DzN, BhN, hPKzMMzR, Hz = N-_ = LT, Hz = N-_LT, SvLDz, DzXBh_BM, DzXBz_LDN, Dz = KBh = M, Hz_Bh = DzN, Dz = KBv = N, hNNz DzK, Dz = XKz = MzR, LhF, MzNSz = Dz, DzK-KzMMzR, GRh_PzBzLKzN, DzXBhDDN, DzXBh_N, h = Dz, Bz = LKzN, SvLLh, ΣTRhBzLKzN, Dz = Kz = MzR, Sv = LDzR, Sv = LzR, Tz_BN-ΣLzX, ΣPz_ = ΣzR, DzXBvN, Hv = Lz, h_WzNz, Hz_ÇBh = DN, Bh_-_
Level 2	BzLK, SvLz, ΣPvΣz, ΣPz_Σz, BhDN, Bv_N, hLz, BvN, SvLDz, KzMz, Bh_DN, hRz, ΣPz_Çz, hDz, ΣPvXz, BzLGz, ΣPvÇz, hLDzR, LvF, ΣPz_Jz, BhN, hLDzN, Bv_DN, Bh_N, BzLzKz, BvDN, DzXBh_, BhLzKzN, ΣPzΣz, ΣPvΣL, Gv_N, DzKhMhN, HzN_LT, Bz_BM, ΣPzJz, Sz_LzR, ΣPvKzR, DzKRh_M, DzXΣRh_T, SvLNzR, hŊzN, hGzLN, ΣPv_ΣzR, SPvÇzR, BzLGŊ, Bv_L, SPvΣzR, hBzHz_S, BhLKN, DJz_LzT, h_BzHz_S, hNzDzK, BhLKzN, DzXLh_KzN, hNzNDzKz, hNDzM DzK, HzNzLT, DzɣBhM, KzLzR, DzXSThBz, ΣPvTZBhN´, ΣPvΣh_, DvNZN, hŊzMDzKz, DvLN, hNDhX, SPz_Σz, DzKRz_M, DzXJzΣhS, DzXGzMzR, DzXBvW, h_WvX, ΣPz_ÇN, BhLGzN, ΣPvSzR, ΣPvTZBhM, BzKN, BzRzN, v_MSzL, LvNPF, Lv_F, Wz_M, DzXBz_LDN, DzKBhM, hNz DzK, LhF, MzNSzDz, SvLh, Tz_BNΣLzX, HvLz, h_WzNz, Bh_
Level 3	P_LK, S_L_, ΣP_Σ_, ΣP_Σ_, P_N, P_DN, P_N, S_LD_, K_M_, ΣP_Ç_, P_DN, P_R_, ΣP_Ç_, _L_N, P_D_, L_F, ΣP_J_, P_L_K_, _LD_N, _L_RN, H_N_LD, _L_, D_ÇP_, P_PM, D_KR_M, ΣP_ΣL, _P_H_S, K_N, _LD_R, ΣP_J_, H_L_, S_L_R, ΣP_K_R, D_ÇΣR_D, S_LN_R, _Ŋ_N, _K_LN, H_LD_N, SP_Ç_R, P_L, SP_Σ_R, DJ_L_D, _N_D_K, D_ÇL_K_N, _N_ND_K_, _ND_M D_K, D_ÇP_M, K_L_R, D_ÇSD_P_, ΣP_DZP_N´, D_NZN, _Ŋ_MD_K_, D_LN, _ND_Ç, SP_Σ_, _R_, D_ÇJ_Σ_S, D_ÇP_F, _F_Ç, ΣP_ÇN, ΣP_S_R, ΣP_DZP_M, P_KN, _MS_L, L_NPF, L_F, F_M, D_KP_M, _N_ D_K, M_NS_D_, _D_, D_PNΣL_Ç, _F_N_, P__
Level 4	P_L, ΣP_Σ_, P_N, S_L_, P_DN, ΣP_Ç_, S_LD_, K_M_, P_R_, _L_N, P_D_, L_F, ΣP_J_, K_N, D_ÇP_, _LD_N, _L_RN, _L_, P_PM, D_KR_M, ΣP_ΣL, _P_H_S, _LD_R, SP_Σ_, H_NLD, H_L_, ΣP_K_R, D_ÇΣR_D, S_LN_R, _Ŋ_N, _K_LN, H_LD_N, SP_Ç_R, DJ_L_D, _N_D_K, _N_ND_K_, _ND_M D_K, H_N_LD, K_L_R, D_ÇSD_P_, ΣP_DZP_N´, D_NZN, _Ŋ_MD_K_, D_LN, _ND_Ç, _R_, D_ÇJ_Σ_S, _F_Ç, ΣP_ÇN, ΣP_S_R, ΣP_DZP_M, P_KN, _MS_L, L_NPF, F_M, D_KP_M, _N_ D_K, M_NS_D_, _D_, D_PNΣL_Ç, _F_N_, P_
Level 5	B, Σ, S, D, O, U, H, L, K, G, W, M, T

The codes in Table 1 and Table 2 already represent the final cluster categories, i.e., after mapping complex codes transitively to existing subcodes (see above HEAPSLAUP→LAUP example) and removing the redundant complex codes. Such a mapping is shown for three levels of *Dachboden* (other words involve fewer mappings, even none, as *Ei*) in Table 3.

Table 3 *Remappings of codes of* 'Dachboden'

Level 2	BhDz→hDz, BhDzN→hDz, BhRz→hRz, BvDNS→BvDN, BvNz→BvN, BvNΣzN→BvN, Bv_Nz→Bv_N, Bv_NzN→Bv_N, BzLGzN→BzLGz, BzLKN→BzLK, BzLKNRhM→BzLK, BzLKz→BzLK, BzLKzN→ BzLK, BzLKŋ→BzLK, BzLzKzN→BzLzKz, DzBhDN→BhDN, DzKBhDN→BhDN, DzKBhDzN→hDz, DzKBhN→BhN, DzKBhN´→BhN, DzKBh_DN→Bh_DN, DzKBh_N→Bh_N, DzKBh_NDN→Bh_N, DzKBvN→BvN, DzKBv_DN→Bv_DN, DzKzMz→KzMz, DzXBhDN→BhDN, DzXBhDzN→hDz, DzXBhN→BhN, DzXBhRz→hRz, DzXBhRzN→hRz, DzXBh_BM→DzXBh_, DzXBh_DN→Bh_DN, DzXBh_N→Bh_N, DzXBvN→BvN, DzXBzLKzN→BzLK, DzXKzMz→KzMz, DzXKzMzR→KzMz, GRh_PzBzLKzN→BzLK, HhLDzN→hLDz, HhLzN→hLz, Hh_BhDN→BhDN, Hh_ÇBhDN→BhDN, HzNzBzLKzN→BzLK, Hz_BhDzN→hDz, KzMzN→KzMz, ShLDz→hLDz, ShLz→hLz, SvLzR→SvLz, SvLz_S→SvLz, hLDzN→hLDz, hLzN→hLz, hLzRN→hLz, hPKzMz→KzMz, ΣPhRz→hRz, ΣPvΣzL→ΣPvΣz, ΣPvΣzRN→ΣPvΣz, ΣPvΣzX→ΣPvΣz, ΣTRhBzLKzN→BzLK
Level 3	D_KP_DN→P_DN, D_KP_D_N→P_D_, D_KP_N→P_N, D_KP_N´→P_N, D_KP__DN→P__DN, D_KP__N→ P_N, D_KP_NDN→P_N, D_K_M_→K_M_, D_K_M_N→K_M_, D_P_DN→P_DN, D_ÇK_M_→K_M_, D_ÇK_M_R→K_M_, D_ÇP_DN→P_DN, D_ÇP_D_N→P_D_, D_ÇP_LK_N→P_LK, D_ÇP_N→P_N, D_ÇP_R_→ P_R_, D_ÇP_R_N→P_R_, D_ÇP__DN→P__DN, D_ÇP__LDN→D_ÇP__, D_ÇP__N→P__N, D_ÇP__PM→P__PM, H_L_N→H_L_, H_N_P_LK_N→P_LK, H_P_DN→P_DN, H_P_D_N→P_D_, H_ÇP_DN→P_DN, KR_P_P_LK_N→P_LK, K_M_N→K_M_, P_DNS→P_DN, P_D_N→P_D_, P_LKN→P_LK, P_LKNR_M→P_LK, P_LK_→P_LK, P_LK_N→P_LK, P_LKŋ→P_LK, P_L_K_N→P_L_K_, P_N_→P_N, P_NΣ_N→P_N, P_R_N→ P_R_, P_N_→P_N, P_N_N→P_N, S_L_R→S_L_, S_L_S→S_L_, ΣDR_P_LK_N→P_LK, ΣP_R_→ P_R_, ΣP_Σ_L→ΣP_Σ_, ΣP_Σ_RN→ΣP_Σ_, ΣP_Σ_→ΣP_Σ_, ΣP_Σ_Ç→ΣP_Σ_, _PK_M_→K_M_, _P_H_S→ _P_H_S
Level 4	D_KP_DN→P_DN, D_KP_D_N→P_D_, D_KP_N→P_N, D_KP_NDN→P_N, D_KP_N´→P_N, D_K_M_→K_M_, D_K_M_N→K_M_, D_P_DN→P_DN, D_ÇK_M_→K_M_, D_ÇK_M_R→K_M_, D_ÇL_K_N→K_N, D_ÇP_DN→ P_DN, D_ÇP_D_N→P_D_, D_ÇP_F→D_ÇP_, D_ÇP_LDN→D_ÇP_, D_ÇP_LK_N→K_N, D_ÇP_M→D_ÇP_, D_ÇP_N→P_N, D_ÇP_PM→P_PM, D_ÇP_R_→P_R_, D_ÇP_R_N→D_ÇP_, H_L_N→H_L_, H_N_P_LK_N→ K_N, H_P_DN→P_DN, H_P_D_N→P_D_, H_ÇP_DN→P_DN, KR_P_P_LK_N→K_N, K_M_N→K_M_, P_DNS→P_DN, P_D_N→P_D_, P_LK→P_L, P_LKN→P_L, P_LKNR_M→P_L, P_LK_→P_L, P_LK_N→P_L, P_LKŋ→P_L, P_L_K_→P_L, P_L_K_N→P_L, P_N_→P_N, P_N_N→P_N, P_NΣ_N→P_N, P_R_N→P_R_, S_L_R→S_L_, S_L_S→S_L_, ΣDR_P_LK_N→K_N, ΣP_R_→P_R_, ΣP_Σ_L→ΣP_Σ_, ΣP_Σ_RN→ΣP_Σ_, ΣP_Σ_Ç→ΣP_Σ_, _PK_M_→K_M_

4.5 DPA: application

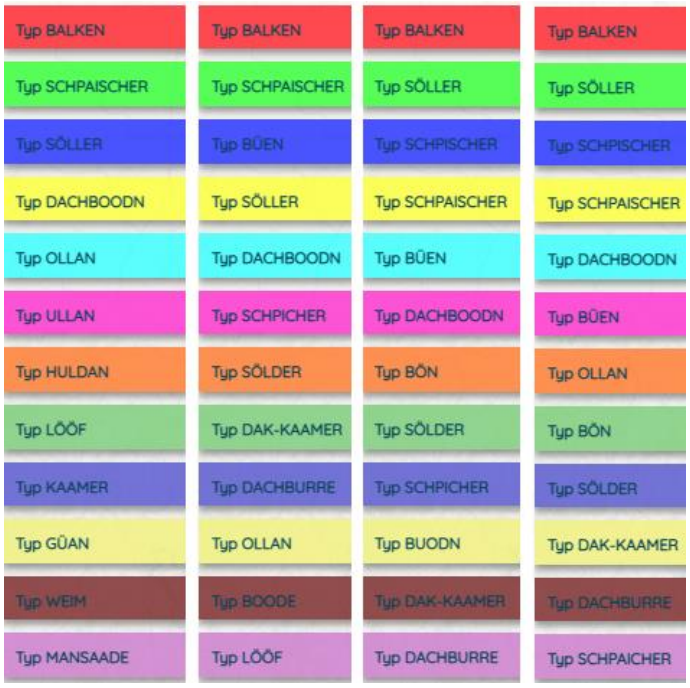
Recall that the purpose of POP clustering is to arrive at a smaller set of POP types in order to obviate cluttered maps and corresponding visual confusion, and to allow for intuitive use and easy interaction with the 'Speaking DMW'. Basically, this is realized by letting the user select the granularity level he/she is interested in, thereby reducing the complexity of presentation and interaction. With a level being set, the POP types to be shown in the legend can then be specified, and the level of detail of the preview map can be determined. This is implemented as follows.

First, the clusters of a level (as well as their POPs) are sorted according to their relevance, i.e., the frequency of their POP *instances*. In the legend, the clusters can then be offered in decreasing order, starting with the most relevant one. Second, each cluster is *termed* to be easily identifiable, not just by the set of POPs it represents (which is shown on mouseover of the cluster field). Unfortunately, it is not possible to come up with an automatic procedure to *generate* a term for a cluster (based on its POPs). However, simply *using* its most relevant POP as its label suffices for this purpose, as the DPA results in distinctive clusters by definition. Third, we assign each of the *twelve* highest ranked clusters of a level one of a set of twelve maximally contrastive colors (correspondingly ranked by their salience). While the number is arbitrary and debatable, the use of colors as opposed to specific symbols is supposed to make the POP types easily recognizable (note that we offer a secondary color palette for people with disabilities in color perception) and to avoid visual confusion. To signify data distribution and diversity on the preview map, we mainly use pie charts (see below). They can be regarded as the only option for the signification of type clusters on our preview maps, both due to the large number of explored places, and to the fact that type diversity per place is frequent for automatically categorised data.

To exemplify the application of the DPA, Figure 7 and Figure 8 show the menus of decreasing granularity level aligned for comparison, each with their clickable, colored, and typed fields corresponding to the clusters. Recall that the up to twelve colored fields are ordered by the number of corresponding variants (further gray-colored fields/clusters not shown here).

Figure 7 *Clickable variant type fields in the legend for the granularity levels of* 'Ei' *(5 to 0)*

Figure 8 *Clickable variant type fields in the legend for the granularity levels of* 'Dachboden' *(5 to 2)*

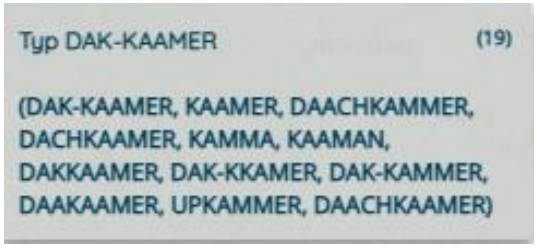


It can easily be seen in Figure 8 that with the "Typ Balken" fields always ranked highest, they correspond to the most frequent clusters (probably different in each case), while the ranks of other types change. For example, type BÜEN (level 4, second column, rank three, darkblue) is ranked higher than the same type on level 3 (third column, rank five, lightblue). This is due to the fact that while

vowel aspects are conflated on level 4, level 3 distinguishes vowel change (diphthongs) from non-change (monophthongs), hence the level-3 differentiation of types BÜEN and BÖN (rank seven, darkorange), both with smaller frequency values.

Figure 9 depicts a single POP type field on mouse-over (grayed out *Dachboden* cluster field on level 2). The cluster is determined by the common code KzMz (of KAMA), and named by the most frequent POP DAK-KAAMMER (the overall number of occurrences of cluster variants given in brackets).

Figure 9 *A level-2 cluster for* 'Dachboden'



Observe that no principles of dialectology are involved in this multi-layered clustering scheme. Apart from the fact that none were available at the time of its development, reaction time of the map interface is essential for user experience (computing level info and corresponding map presentation of the data on level choice). It can be expected that any more sophisticated scheme (for example, considering location aspects of variants for categorization) would slow down performance noticeably.

5. Aspects of Preview maps

There are a number of requirements regarding content (presentation), interaction, and (web) technology to be considered in implementing the map interface. In the following, these requirements and their corresponding solutions are discussed.

5.1 Content (presentation)

At the core of the DMW system, there is the mass of diverse dialect(ological) data, and the requirement to taylor its presentation to different *user* classes (lay people and experts). More specifically, there are object and meta data of exploration and analysis, some of which are relevant for the map interface. Primarily, the preview maps are supposed to show the analysis data for some exploration item (a RWL), and to provide the facility to hear the correspondingly cutted audios (the observants of the RWSes) of the informants at some explored place.

Within this basic functionality of the "Speaking DMW", one has to distinguish the audio-visual presentation of variants of words via the POP mechanism, and the possibility to select and hear (multiple) Wenker sentences. In the map interface, *word maps* and *Wenker maps* are offered for this distinction, respectively (see Figure 10 and Figure 11). Both figures show the removable central half-transparent map query menus and the popup-windows containing the wav/mp4 players at a certain mouse-clicked place. The map types are different in that Wenker maps lack the legend of word maps. Also, while a word map popup contains players for the variants of each person (i.e., audio on demand), playing the Wenker sentences starts on click, and as a looped sequence of all wavs/mp4s of the selected Wenker sentences, and of all persons at that place (navigation backward and forward is offered).

Figure 10 *Word map of* 'Kette'

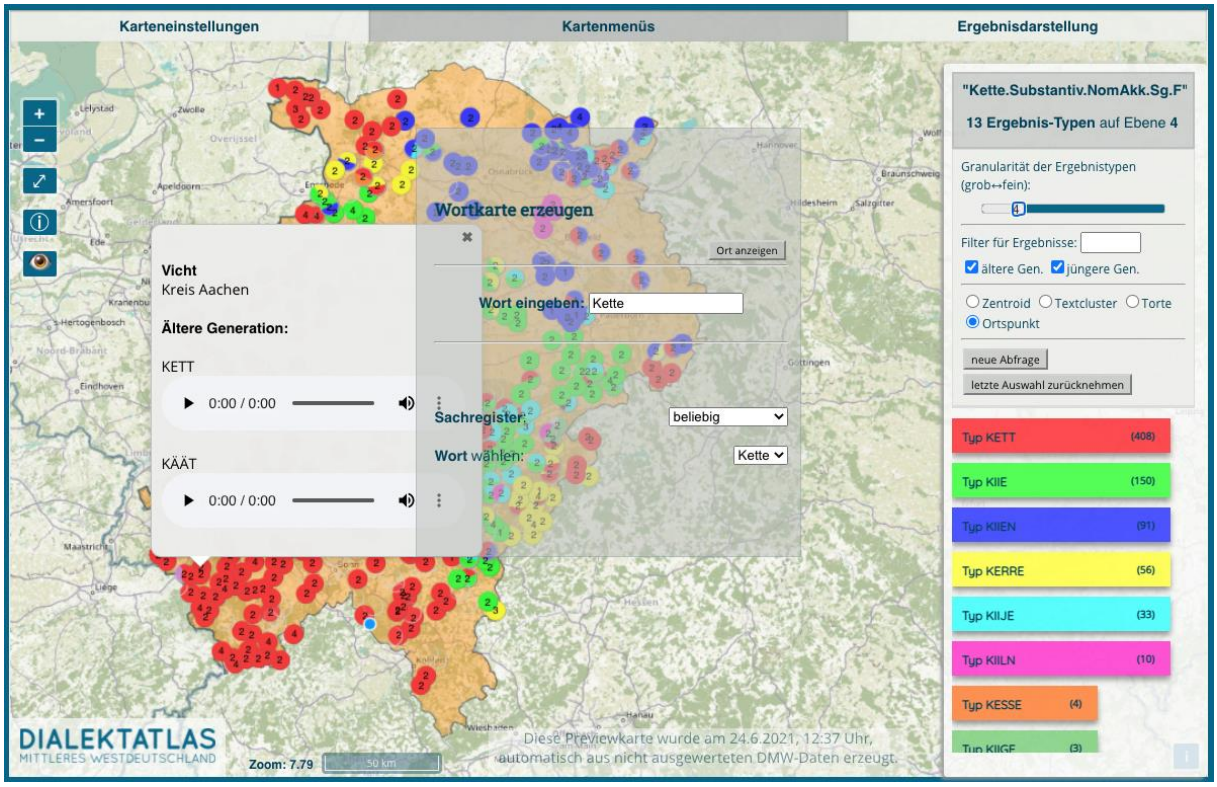
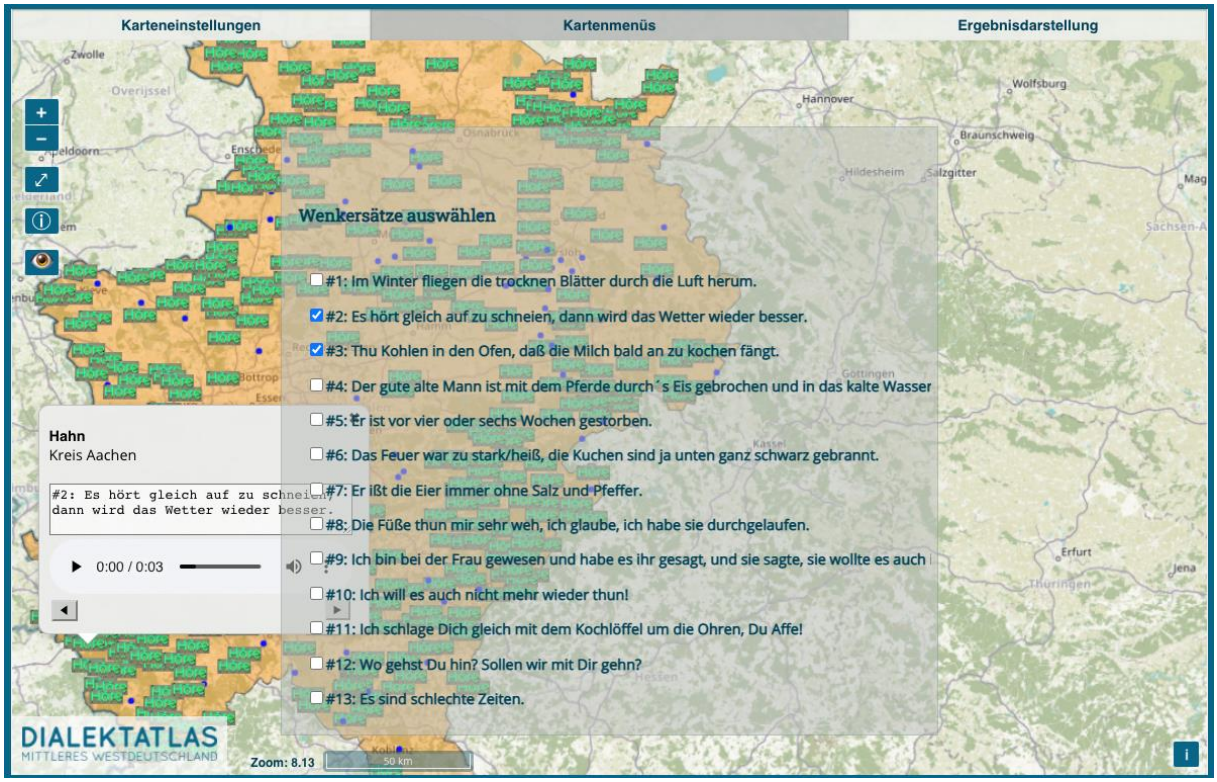
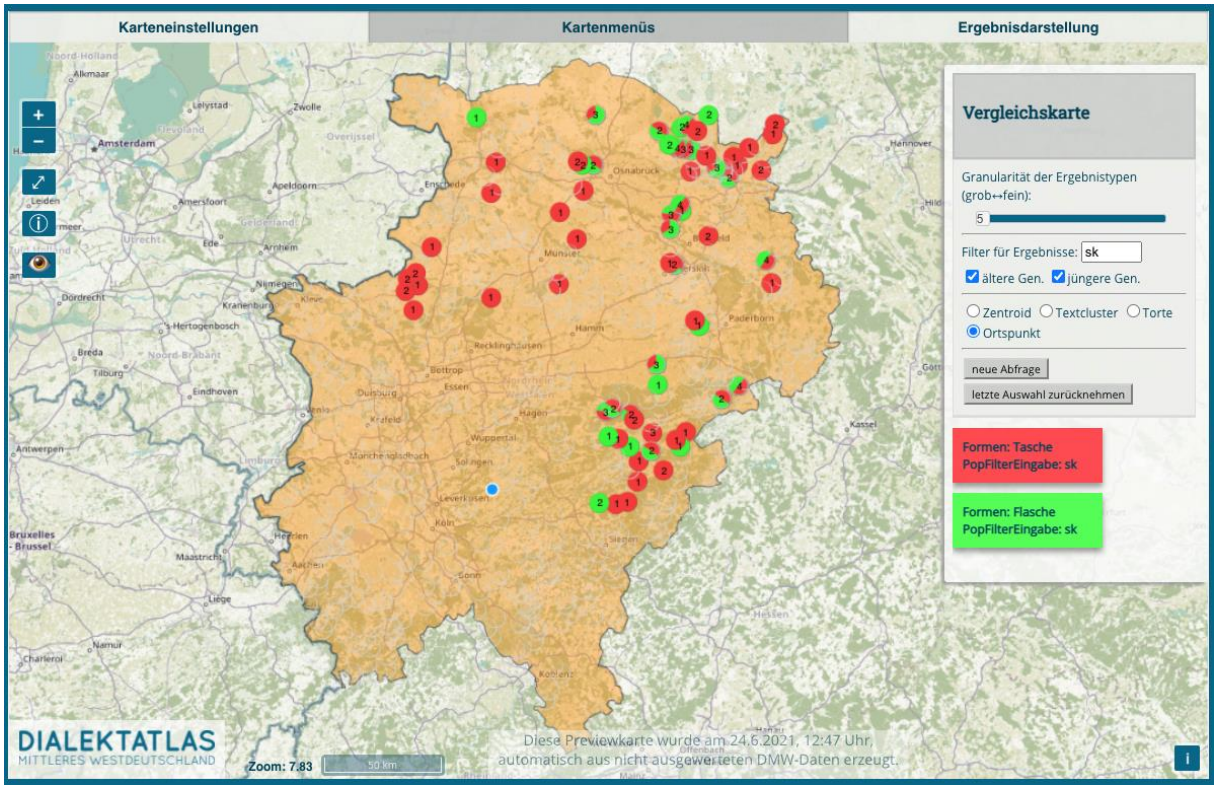


Figure 11 *Wenker map of selected Wenker sentences*



It is also possible to construct several word maps at the same time, and to compare the corresponding distributions on a so-called *comparison* map ("Vergleichskarte"). Figure 12 shows such a display, in which (only) occurrences containing *sk* of *two* words (*Tasche, Flasche*) are mapped. In the expert version, phenomenon-related analysis data will be presented on corresponding phenomena preview maps. In the legend of a comparison map, the variant type fields do not show POP(type)s, but the specifications of the query for the corresponding map-to-be-compared (i.e., the list of option choices made).

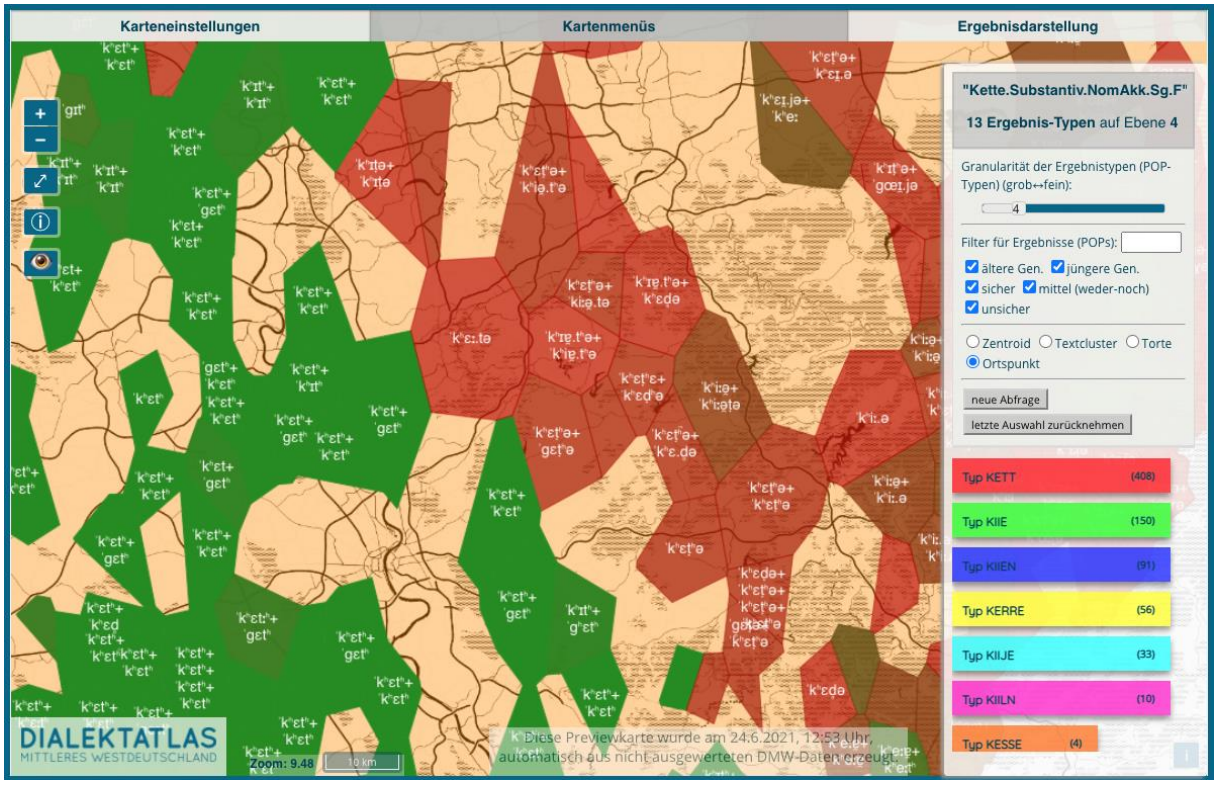
Figure 12 *Comparison map*



The currently available dialectometric type is a reference point map, which codes distances between variants of a clicked-on reference cell and each other cell as colors on a linear green-red scale (close to distant).[20] There are two mechanisms to ensure adequacy of this scale for the probable case of very different variants (where red would be the dominant color). First, the sorted domain of values is cut off at some point, and outliers receive the maximum value ("clamping"), which better reflects the differences in the proximal range of values, while only conflating different distant values. Second, reference point comparison is performed for the places of the browser window's content, i.e., the currently "visible" area. Different from a global process, this better mirrors the finer distinctions of closer related variants in zoomed-in situations. Figure 13 displays such a reference point map for *Kette*, showing "regions of similarity".

[20] Since each place typically has more than one, and potentially different, variants, the smallest value (i.e., of the closest variants) is selected. As to the empty/transparent cells, data for the corresponding place are either not yet analyzed, or have been sorted out as invalid or irrelevant.

Figure 13 *Reference point map*



In general, meta data are used to restrict data presentation, for example, to some age group, or to some confidence-of-transcription level (the POP mechanism can be regarded as another case in point, with the index terms representing similarity data about the variants).

For the presentation of our mass data of variants, it is necessary to guarantee the *perceptibility* of data variation and distribution. As to *data variation*, its confusion is prevented by the POP mechanism, with which different similarity-based clusters of the data can be computed (and selected) on six levels of granularity with the DPA described above. By starting with level 5, this corresponds to having a coarse overview at the beginning.

The perception of *data distribution* is facilitated by four different kinds of *locational clusters* and their distinctive symbolization on preview maps, and by offering four corresponding options of presentation. All of them share the use of *color* for the symbolization of variant types, as the use of specific symbols would easily lead to visual cluttering for higher numbers of types (and is therefore forbidden in the face of our data variety).[21]

The first option is *place*-oriented data clustering (“Ortspunktdarstellung”), according to which the data of a place (from up to four persons) are clustered and presented as a small, point-like pie chart (the pie chart showing corresponding color portions of the data distribution, as well as the number of persons), as in Figure 1, Figure 10 and Figure 12.

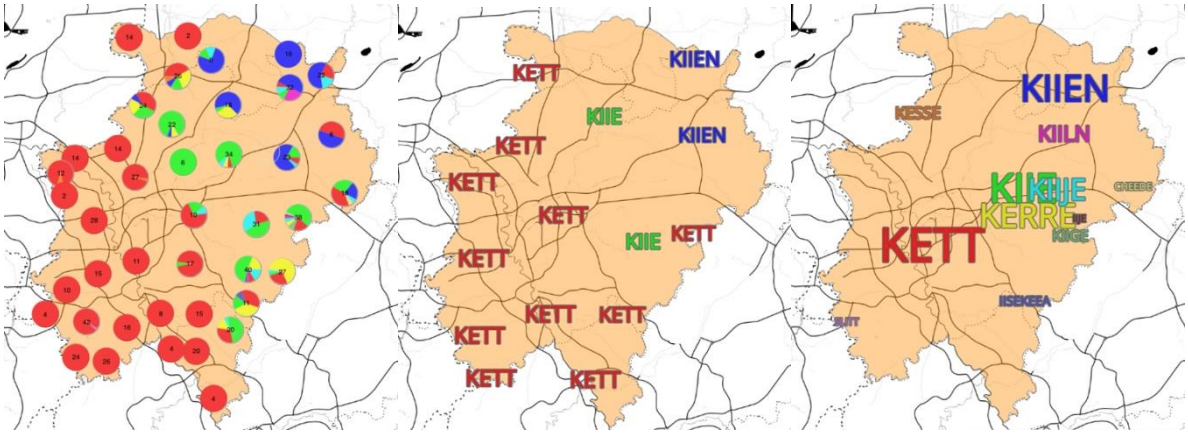
[21] The perception of the (up to twelve) colours on word maps used to distinguish types is enhanced by the selection of high-contrastive ones for this purpose. This use of colours therefore is a basic feature of the DMW system, and marks our secondary colour palette as a stopgap for people with disabilities in colour perception.

The second option uses Open Layers' facility of *cluster maps*, by which data from multiple places are automatically clustered according to some scheme (grouping via distance to some raster points, whose application on different zoom levels leads to cluster sets of varying granularity). Here, the same, but bigger, pie charts are employed ("Tortendarstellung"), only that the data are gathered from the places of the respective cluster (see Figure 14.a).

While such cluster maps combine detailed information presentation with some scheme of visual abstraction, they still might be confusing, because the color-type relationship must be kept in mind or looked up, and may distract from the gist of the data distribution. Hence, the third option of *text cluster maps* allows to only show the main (POP) type of each cluster as colored text, which allows a constricted, but intelligible, raw view on the data distribution at a certain zoom level (see Figure 14.b). In case of prominent secondary types (more than 40% portion), they are each displayed directly below the primary ones.

What is still missing, is a *wholistic* view on data distribution that abstracts from different-zoomlevel variation. For this reason, the fourth option clusters *all* place data of a variant type, computes the spatial center ('centroid') of the cluster ("Zentroiddarstellung"), and presents the colored POP-type text of the cluster at that point (text size varying with the number of variants), as in Figure 14.c. This allows to display the relative position of different variant types' bulks of variants even in cases where they widely overlap.[22]

Figure 14 a. Cluster b. Text cluster c. Centroid



Aside from these word-map means of facilitating distribution perception, reference point maps can be used to recognize fine-grained differences and similarities in the data.

Another means to allow for better perceptibility of the data is *data selection for presentation*. The map interface offers three facilities to do that. The first is *momentary variant type selection* by the use of Open Layers' *heat maps*. They are generated if a user does a mouseover on a variant/POP type field in the legend. In a heat map, the places of the selected POP type are foregrounded with a blurred color scheme (red core, yellow inner border, green outer border). In the case of neighbouring places, this results in a confluent, intensified red coloring, highlighting that region (and therefore marking relative

[22] Again, this computation is window-dependent. Especially when zooming out, therefore, "centroid" view must be re-selected.

distribution). At the same time, all other information is hidden (compare Figure 15 and Figure 16). On mouseout, everything is restored.

Figure 15 *Heatmap of POP type KIIE of* 'Kette' *(level 4)*

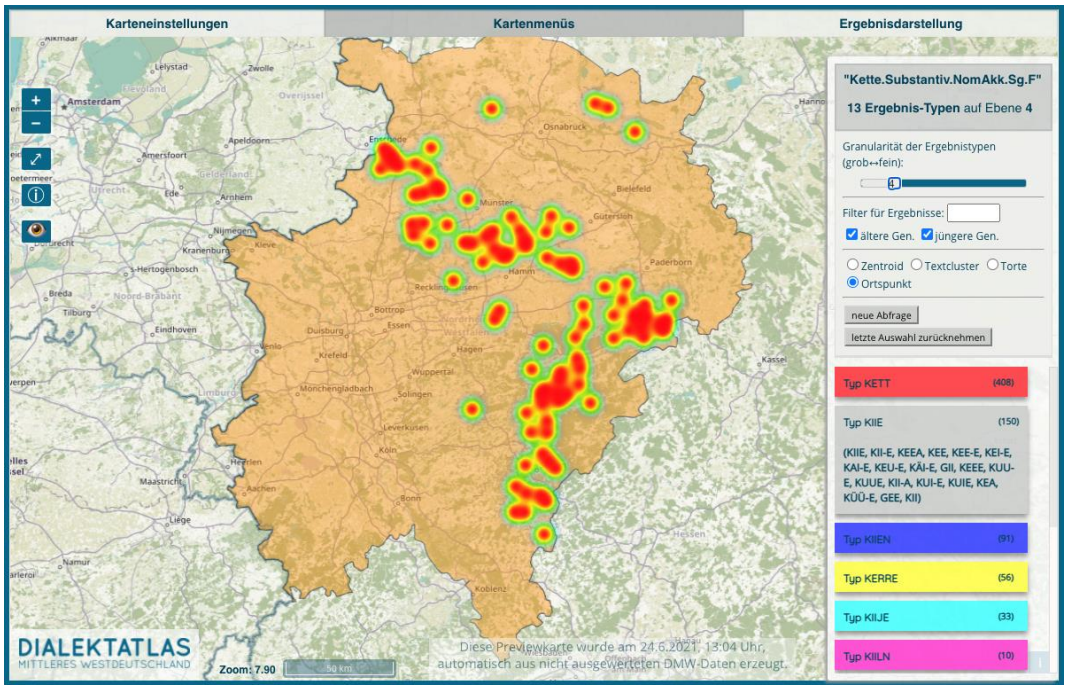
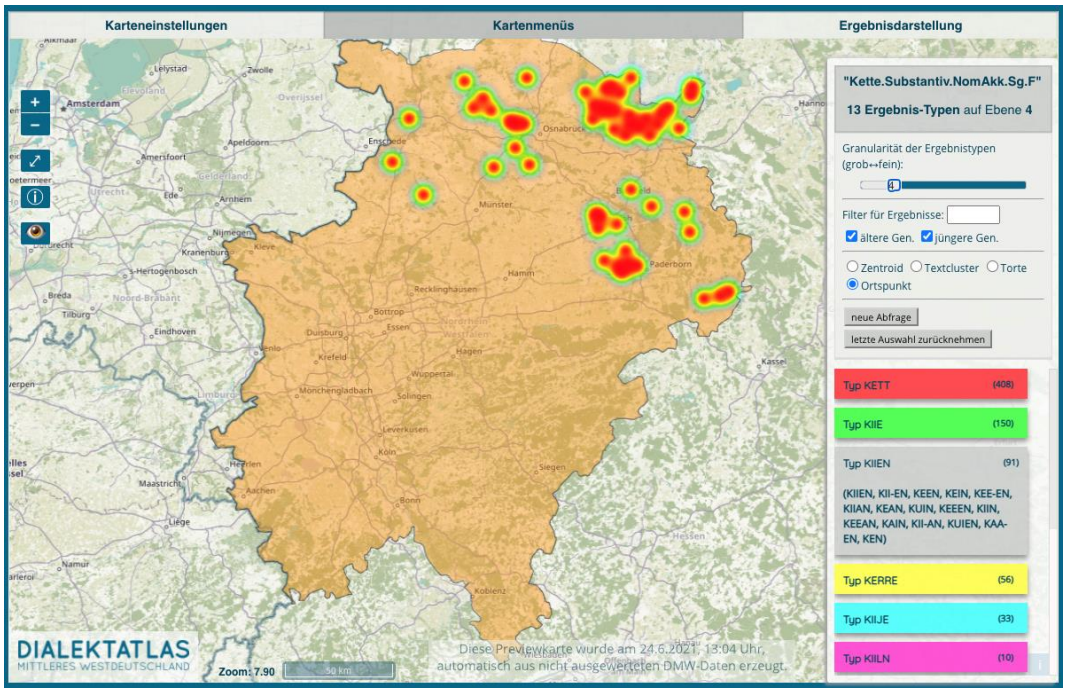


Figure 16 *Heatmap of POP type KIIEN of* 'Kette' *(level 4)*

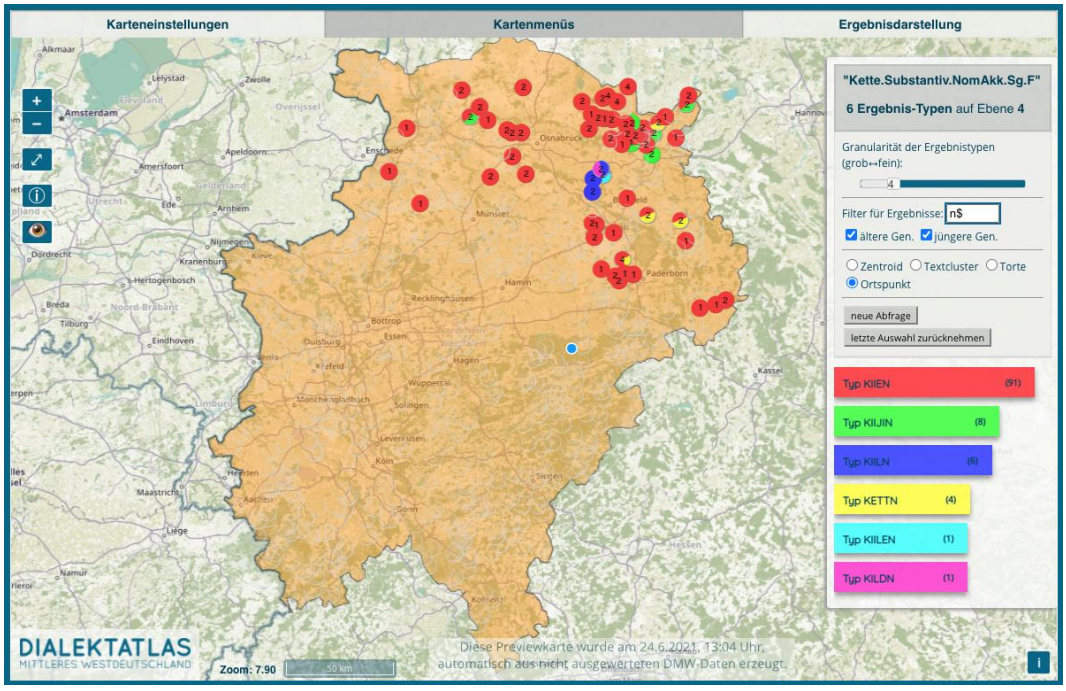


The second is *permanent variant type selection* by double-clicking a POP field in the legend, given some granularity level. In this case, presentation is restricted to the data corresponding to the current-level POP type in question. This is especially useful for the investigation of a certain class of variants, before

lowering the granularity level (again). The selection remains active until it is explicitly taken back by clicking on “letzte Auswahl zurücknehmen” (“Undo last selection”).

The third aspect is *filtering*. This is done by entering regular expressions in the corresponding field of the legend, which are used to filter the set of POPs before (re)application of the DPA. Filtering can be performed on any level of granularity (see Figure 17 for a *Kette* word map with the filter set to “ends with a ‘n’” on granularity level 4).

Figure 17 *Word map* Kette *restricted to variants ending with* ‘n’



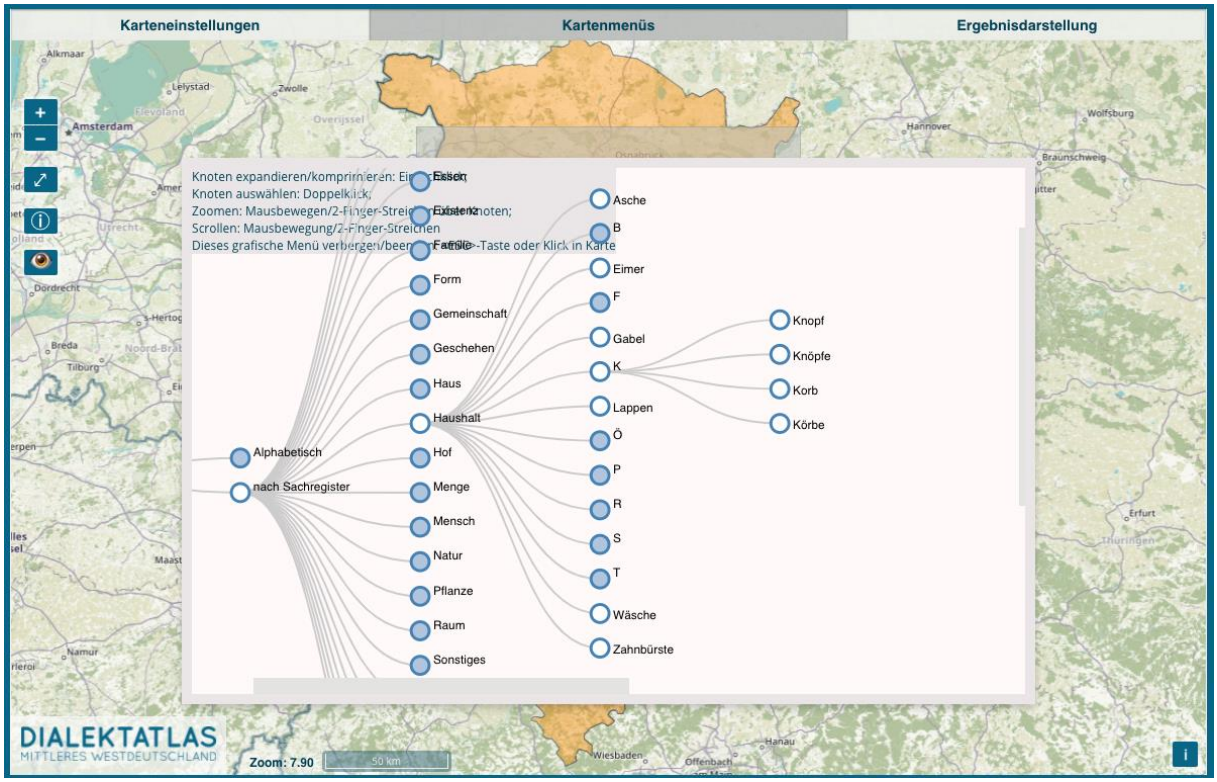
Needless to say that these means of dealing with the perceptibility of data variation and distribution can be used in combination. For example, by setting a granularity level, clicking on a POP type field, setting a lower granularity level, and finally entering a filter expression. By undoing last actions and entering a different part/path of the possible options, this procedure corresponds to *exploring* the distribution of some RWL’s variants.

5.2 Interaction

There are some requirements regarding user interaction in the map interface. First of all, this includes aspects of *usability* and ergonomics, given the two basic classes of users (lay people, experts): especially, an intuitive, easy handling of the interface, which should not only be (visually) parsimonious, but also self-explanatory at best. Then, the map interface should allow *different perspectives and choices*, as well as individual preferences for one or the other. Finally, interaction should also be *explorative*. This requirement for visualization applications can be described by the following steps of visual analytics (Keim et al. 2010): Provide *overview* (most important things first); offer *intuitive options* (for example, zooming/filtering, restricting, focusing, navigating); provide *details on demand*; show *relationships*; *easy switching* between aspects to be displayed. We tried to meet these needs as follows.

First, there is no distinction between a query page (specifying the type of data to be presented) and a map page (which was the case in the small precursor project SiSAL, see Solau-Riebel and Vogel 2013-2016), simplifying interaction. Instead, the centered query menus –being the currently focused elements of interaction– are always displayed *on top* of the geo-centered map and data layers (even with a graphical selection menu on top of the standard query menu, see Figure 18). Likewise, removing the top level corresponds directly to falling back to the next important lower level, or established state of presentation. Interaction is further facilitated by offering additional shortcuts to close menus and popups.

Figure 18 *Graphical selection menu based on a D3-style hierarchy*



Second, the *complexity* of the menus is simple by default, but changes on demand. For example, our basic query menu of word maps contains only three options (entering a word, selecting an ontological domain, and selecting from a word list). However, there is a check box allowing for the display of more options (selecting from a lemma list, choosing a part of speech (which will show pertinent suboptions), and choosing from a graphical menu). The Wenker menu simply lists all wenker sentences to choose from with check boxes, yet entering parts of words acts as a filter and immediately leads to a reduced check list. In general, menus of a certain type not only (dis)appear on mouse click, but also on certain mouseovers or keyboard entries, where appropriate. Correspondingly, users may develop their own preferences how to interact with the system.

Third, we follow the principle of successive refinement of a query by offering ontological or linguistic categories to select from (in part, hierarchically). Technically, this is supported by automatically restricting *all* options to the remaining compatible values after some selection ("self-restricting options"). To further improve performance, specifying a query is detached from database

lookup, the latter being triggered only if some condition is met (for example, “only one item left in the list of available words” on a word map). Query specification is therefore a light and quick process, which, with the possibility of undoing the last choice made (by clicking “letzte Auswahl zurücknehmen”), can be performed exploratively, too.

Fourth, *explorativity* is at the heart of the map interface (as has already been shown). For example, both the query selection and the granularity mechanism start with an overview; mouse over a POP field only shows the corresponding places (as a heat map) of that variant; detailed information about the data represented by a pie chart can be requested and is presented in a popup (percentage of the variant types at that location, with absolute numbers of places and persons); the relationship of similar pronunciation is made visible on reference point maps; setting a different background map or using a different kind of map is only one click away, and likewise –apart from the four data distribution display options–, zooming in or out in cluster maps instantaneously yields different perspectives on the data. A final aspect of comfortable explorativity is the use of comparison maps, where selected variants of possibly different words can be easily specified and compared.

There is a qualification regarding usability, however. Although our interaction scheme works well ergonomically, both with respect to content and query, it is far from being self-explanatory. Instead of abandoning our interaction principles, however, we simply decided to offer videos of the basic functionality at the entry to the map interface, and an elaborate help menu. This design choice for a steeper learning curve, but long-term interaction benefit, corresponds to a preference for multiple-time visiting interested users of the map interface.

5.3 Technical aspects

The basic technical requirements of our map interface can be summarized as follows: to be able to automatically generate dynamic preview maps with state-of-the-art web software components; to have a responsive application, both with regard to query and presentation; and to have a persistent implementation to some extent.

We currently use a Parcel-bundled Open Layers 5.3.0 ES6-style Javascript application featuring JQuery(-UI) and D3 npm-organized modules[23] that accesses a MySQL database for text data, and provides links to the scientific cloud Sciebo for audio data. For tuning the selection, and presentation, of values from lists for certain input fields, we use the *awesomplete* autocompletion package by Lea Verou. The chores of linguistic data preparation for the query menu options (e.g., for hierarchical graphical menus like the one in Figure 18, but especially for the phenomena-related options of the expert version not described here) are performed separately with Node.js. As to map generation, we needed to use Voronoi cells of the explored places for our reference point maps (to provide for colorable areas). These cells (which are generated in QGIS® and imported as kml data) are also used to react tolerantly on place clicks in word maps. For fluid interaction, the map interface is heavily event-driven, minimizing clicking effort (for example, using mouseover for the initiation of some action, or

[23] Not loading such components via – potentially defunct – links to software sites guarantees availability of the used software components, and hence, some persistence of the map interface. For long-term persistence, we may use a Docker-style implementation of the DMW system.

automatically focusing on an input field if its parent is foregrounded). In addition to that, database calls are minimized, at least for the present map types.

6. Conclusion

Starting with an overview of the data and work flows, and of transcription, in the DMW project, this article described the algorithmic and technical aspects of the project's preview word map generation. It was shown how the problem of massive dialect data variation can be overcome by using a multi-level Soundex-like indexing scheme of the transcribed observants, ultimately achieving effective, digital, dynamic, interactive visualization. In the course of this, different aspects of preview word maps were explained, and examples for their explorative use in what is called "Speaking DMW" or "Dynamic atlas maps of the DMW project" were presented.

7. Acknowledgments

I would like to thank Aynalem Misganaw (from the center of information and media technology, ZIMT, of the University of Siegen) for her substantial database-related work, which made all this possible. She implemented all front ends except the map interface. I am grateful to Hans Goebl for fruitful discussions. I also thank Ambra Ottersbach and Nadine Wallmeier for helpful comments on versions of this paper. Finally, thanks go to all (student) colleagues whose feedback has helped shaping the DMW system to what it is now.

8. References

Bieberstedt, Andreas, Ruge, Jürgen and Schröder, Ingrid 2016: Hamburger Transkriptionskonventionen. In Bieberstedt, Andreas, Ruge, Jürgen and Schröder, Ingrid (eds.): *Hamburgisch. Struktur, Gebrauch, Wahrnehmung der Regionalsprache im urbanen Raum*. Frankfurt am Main u. a. (Sprache in der Gesellschaft, 34), 421–428.

Carstensen, Kai-Uwe, Spiekermann, Helmut, Tophinke, Doris, Vogel, Petra M. and Wich-Reif, Claudia 2020: Zur Methodik des Dialektatlas Mittleres Westdeutschland (DMW). *Korrespondenzblatt des Vereins für niederdeutsche Sprachforschung* 127: 107–114.

Draxler, Christoph and Jänsch, Klaus 2004: SpeechRecorder – a Universal Platform Independent Multi-Channel Audio Recording Software. In: *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, 559–562.

Draxler, Christoph and Jänsch, Klaus 2019: SpeechRecorder.
URL: https://www.bas.uni-muenchen.de/Bas/software/speechrecorder/ (last accessed 04.06.2022).

Gehrke, Gero, Kuhmichel, Katrin, Sauermilch, Stephanie and Wallmeier, Nadine 2020: Dialektatlas Mittleres Westdeutschland (DMW) – Methodik, Akquise, Exploration und Analyse. *Niederdeutsches Wort*. 60: 7–33.

Goebl, Hans 2010: Dialectometry and quantitative mapping. In: Lameli, Alfred, Kehrein, Roland and Rabanus, Stefan (eds.): *Language and Space. An International Handbook of Linguistic Variation, vol. 2: Language Mapping* (Handbücher der Sprach- und Kommunikationswissenschaft [HSK] 30.2.), Berlin: de Gruyter 2010; 1st part: 433-457 (text), 2d part (maps): 2201–2212.

Haimerl, Edgar 2005: Taxierungsalgorithmen. In: *Quantitative Linguistics; an International Handbook*. Eds. Köhler, R. Altmann, G. De Gruyter. 532–547.

Keim, Daniel, Kohlhammer, Jörn, Ellis, Geoffrey and Mansmann, Florian (eds.) 2010: *Mastering the information age: solving problems with visual analytics*. Goslar: Eurographics Association.

Lameli, Alfred, Kehrein, Roland and Rabanus, Stefan (eds.) 2010: *Language and Space. An International Handbook of Linguistic Variation, vol. 2: Language Mapping* (Handbücher der Sprach- und Kommunikationswissenschaft [HSK] 30.2.), Berlin: de Gruyter.

Ruge, Jürgen 2019: *Hamburger Transkriptionskonventionen: lautnah – lesbar – auf Modifikation ausgelegt* [Hamburgian transcription conventions: close to phonation – readable – open to modification]. Talk given on the DMW Workshop „Methoden der Transkription und Transliteration dialektaler Daten“ [“Methods of transcription and transliteration of dialect data”] (Münster, 09.08.2019).

Schmidt, Jürgen Erich, Herrgen, Joachim and Kehrein, Roland (eds.) 2008ff.: *Regionalsprache.de (REDE). Forschungsplattform zu den modernen Regionalsprachen des Deutschen*. With the collaboration of Dennis Bock, Brigitte Ganswindt, Heiko Girnth, Simon Kasper, Roland Kehrein, Alfred Lameli, Slawomir Messner, Christoph Purschke, Anna Wolańska. Marburg: Forschungszentrum Deutscher Sprachatlas.

Solau-Riebel, Petra and Vogel, Petra M. 2013-2016: Siegerländer Sprachatlas (SiSAL). URL: http://www.mundart.sisal.uni-siegen.de (last accessed: 01.05.2021).

Spiekermann, Helmut H., Tophinke, Doris, Vogel, Petra M. and Wich-Reif, Claudia (eds.) 2016ff.: Dialektatlas Mittleres Westdeutschland (DMW). Siegen: Universität Siegen [URL: https://www.dmw-projekt.de/].

Wells, J.C. 1997: SAMPA computer readable phonetic alphabet. In: Gibbon, Dafydd, Moore, Roger and Winski, Richard (eds.): *Handbook of Standards and Resources for Spoken Language Systems*. Berlin and New York: Mouton de Gruyter. Part IV, section B. 684–732.

Wieling, Martijn, Nerbonne, John and Baayen, R. Harald 2011: Quantitative Social Dialectology: Explaining Linguistic Variation Geographically and Socially. *PLoS ONE*, 6(9): e23613. doi:10.1371/journal.pone.0023613.

Wilz, Martin 2005: *Aspekte der Kodierung phonetischer Ähnlichkeiten in deutschen Eigennamen*. Magisterarbeit an der Philosophischen Fakultät der Universität zu Köln.

DOI: https://doi.org/10.14232/wpcl.2022.7.3

Tools for data processing and visualization in the project VerbaAlpina

Beatrice Colcuc and Florian Zacherl[1]

1. Project overview

Since the beginning of the project in 2014, VerbaAlpina (https://www.verba-alpina.gwi.uni-muenchen.de/) has been a research project which represents a connection between linguistics and informatics from the perspective of the digital humanities. VerbaAlpina is a cooperation between the Institute of Romance Philology and the Center of Digital Humanities of Munich University (http://www.itg.lmu.de) and combines linguistics and information technology. An extensive, multilingual research environment with several functional areas has been created by using up-to-date media technology. Thus, a prototype for the transfer of traditional geolinguistics into digital humanities has been developed.

VerbaAlpina has always emphasised the importance of not considering the research activities solely from an internal perspective but inserting the project into a broader framework, by appropriating innovative methods offered by digital technology and, in particular, by developing these digital tools and methods further. In this way, VerbaAlpina has always envisioned the project from the perspective of FAIR thinking, even before the four cardinal principles of modern and digital research were formulated in 2016 by Wilkinson and Dumontier et al. (2016).

1.1 The area under investigation: the Alpine region

VerbaAlpina[2] is a research project based at Munich University which has been funded by the *DFG* (German Research Foundation) as a long-term project since 2014. VerbaAlpina seeks to investigate the linguistic and cultural area of the entire Alpine region from a transnational perspective. The area under investigation corresponds to the perimeter of the Alpine Convention (an international treaty between all bordering states of the Alpine region)[3] and covers a surface of 190,600 km^2. Thus, different countries are part of the project, namely Germany, Austria, Switzerland, Italy, France, Liechtenstein, Slovenia and Monaco.

From an ethnographic and topographic point of view, the Alpine area shows a constant homogeneity throughout the territory. From France to Slovenia livestock breeding, milk and cheese production, and also flora and fauna are common to the whole mountain region. However, when looking at the linguistic composition of the Alps, great heterogeneity emerges. In this area, varieties from three language families are spoken, namely Romance, Germanic and Slavonic. This plurality manifests itself not only

[1] We would like to thank our colleague Markus Kunzmann with whom we held our talk during the workshop. He contributed significantly to the conception and the contents of this paper.

[2] The complete name is “VerbaAlpina. Der alpine Kulturraum im Spiegel seiner Mehrsprachigkeit” (VerbaAlpina. The Alpine cultural region reflected through its multilingualism; Krefeld/Lücke 2014-).

[3] https://www.alpconv.org/en/home/.

in the number of standardised national languages (i.e. French, Italian, German, Romansh and Slovenian) spoken and written in the territory but above all in the various local varieties which are very well preserved in a large part of the region. The fragmentation of the Romanic area is more pronounced compared to the Germanic zone, and the latter, in turn, is more fragmented than the Slavonic zone. Alpine dialects are historically primary ones.[4] This means that they have developed before standardised languages could emerge and cover these as umbrella languages (cf. Krefeld 2020).

For VerbaAlpina it is fundamental to differentiate between the large (in terms of the number of speakers) standard languages spoken and written in the Alpine region (Italian, French, German, and Slovenian), the smaller standardised languages (Romansh, Ladin, and Friulan) and the language varieties of the dialect continuum. VerbaAlpina is mainly interested in the diatopic variation of the Alpine region. In other words, VerbaAlpina considers only dialectal words as its primary data. During the third stage of data processing, namely during the typification, dialectal words are grouped and matched with lemmas from standard dictionaries as will be shown below.

1.2 Research aims

As a research project, VerbaAlpina pursues several aims that concern both linguistic and IT (or infrastructural) aspects.

On the one hand, VerbaAlpina aims to investigate the vocabulary of the dialect varieties of the entire Alpine region in a selective (dealing with different semantic domains) and analytical (processing data in different steps) way. The project was planned in three stages: During the first stage (from October 2014 to October 2017) the focus was on vocabulary from the field of Alpine pasture farming, especially milk processing. The second stage (from November 2017 to October 2020) was dedicated to the vocabulary relating to flora, fauna, landscape formations, and weather. The subject of investigation of the third stage (from November 2020 to October 2023) is the vocabulary of modern life in the Alps, especially ecology and tourism. This approach helps to detect and to underline differences and similarities not only between linguistic varieties within the same language family but especially between linguistic varieties of different language families. One of the aims is therefore to recognize connections regarding the etymology of the individual dialectal words and to reconstruct their historical linguistic paths. Many words share a common etymology, even if this cannot be seen anymore at first sight: e.g. German *Butter*, French *beurre* and Italian *burro* are immediate developments from the Greco-Latin *butyru(m)*. By adopting this approach, VerbaAlpina overcomes the traditional boundaries of nation-states carrying out broader geolinguistic research.

On the other hand, VerbaAlpina is engaged in setting up a portal by using modern media technology: documentation, tools for data collection and processing, and collaborative development are aspects on which VerbaAlpina works.

In order to create a solid network of cooperation in the Alpine region, VerbaAlpina can count on numerous project partners. These include sister projects, cultural institutions in the Alpine region, and organisations involved in scientific research in the fields of language and information technology. The list of partners can be viewed at the following address: https://www.verba-alpina.gwi.uni-muenchen.de/en/?page_id=185&db=202.

[4] An exception is represented by the Walser community.

2. Data gathering and processing

VerbaAlpina works on a lexical level. Specifically, VerbaAlpina is about recording data from printed or digital sources in a structured way in a database. The data VerbaAlpina gathers and analyses derives, on the one hand, from printed linguistic atlases and geo-referenced dictionaries from the past one hundred years. For instance, VerbaAlpina deals with the *Sprach- und Sachatlas Italiens und der Südschweiz* (Jaberg and Jud 1928–1940), the *Atlas linguistique de la France* (Gilliéron and Edmont 1897–1900) and the *Dicziunari Rumantsch Grischun* (DRG; De Planta et al. 1938ff). VerbaAlpina also disposes of digital data from partner projects (e.g. the Bavarian Dialect Database which contains the surveys for the language atlas projects from the Bavarian Language Atlas).[5] Moreover, linguistic data acquired from atlases and dictionaries are supplemented through crowdsourcing. The crowdsourcing platform was designed within the project and directly addresses speakers of the Alpine dialects. By this new collection of current linguistic material, inconsistencies between the existing sources shall be evened out, gaps shall be eliminated and obsolete designations shall be marked as such. In this approach, the Alpine region can also be analysed from a diachronic perspective (cf. Krefeld and Lücke 2020a).

Concerning data processing, VerbaAlpina has to face the challenge that consists in the lack of uniformity of data from different sources as the data is not structured in the same way. In the VerbaAlpina portal, sources from different research traditions (Romance studies, German studies, and Slavonic studies) which were completed at different historical stages of dialectological research are brought together. To unify the linguistic material, data from printed sources first has to undergo a transcription process. The fact that data is so heterogeneous implied that there was a need for appropriate tools to satisfy the two opposing principles of reliability to the source and the easy comparability of the linguistic material.

2.1 Transcription

The linguistic material is entered into the MySQL relational database through a transcription system based only on ASCII (American Standard Code for Information Interchange) characters. The processing of linguistic attestations is therefore possible with any keyboard, by any user and at any time. The system used by VerbaAlpina, following the terminology used by the *Thesaurus Linguae Grecae* (TLG; Brunner 1972-), which first developed this system in the early 1970s for the electronic recording of ancient Greek texts, has been named “Beta Code”.

This system establishes that each linguistic character contained in an atlas or in a dictionary used by VerbaAlpina corresponds to a combination of ASCII characters.

In order to follow the two main principles mentioned above, which require the reliability of the source and the comparison between data, VerbaAlpina has decided to reproduce linguistic material graphically in two different ways:

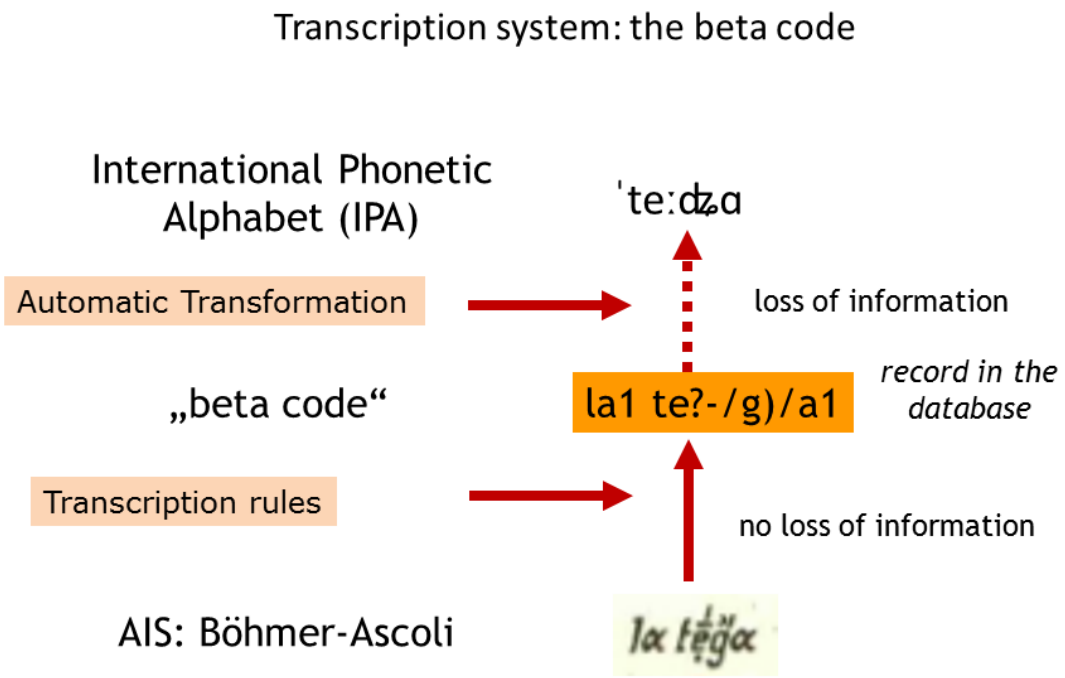
(1) Original transcription: As already mentioned, VerbaAlpina deals with sources that belong to different historical research periods. Therefore, it is necessary to respect the original transcription. Nevertheless, due to technical reasons, it is impossible to keep certain conventions unchanged, especially when several diacritical symbols overlap a letter. In VerbaAlpina, using the Beta Code, this vertical system of characterization of a letter is transferred to linear sequences of characters.

[5] For the complete list of atlases and dictionaries cf. Colcuc/Krefeld (2020).

For the beta encoding, ASCII characters that resemble (as far as possible) the original characters are used. This graphical rendering of the original transcriptions using Beta Code avoids possible loss of information.

(2) IPA: Using specific substitution routines, all Beta Codes are transferred to IPA characters. This is crucial for comparability and user-friendliness. The transformation of a sequence of characters into IPA can sometimes lead to a slight loss of information. In fact, the opening of vowels in IPA is much less precise than in transcription systems such as Böhmer-Ascoli or Theutonista.

Figure 1 *The transcription system of VerbaAlpina*



As briefly mentioned above, VerbaAlpina also works with data from digital databases and with data collected directly from speakers via crowdsourcing. Digital data from partner databases is automatically imported into VerbaAlpina and therefore does not need to be transcribed via the transcription tool. Some atlases are converted into beta code before being imported, others are imported in their beta code so that they can be further transformed into IPA. Other sources are directly imported in IPA transcription. We also deal with sources that are imported in an existing Unicode representation, but in these cases an IPA conversion can still be carried out. In principle, VerbaAlpina has no IPA representation for data from crowdsourcing because it is not possible to convert this type of attestations automatically.

2.1.1 Transcription rules and transcription tool

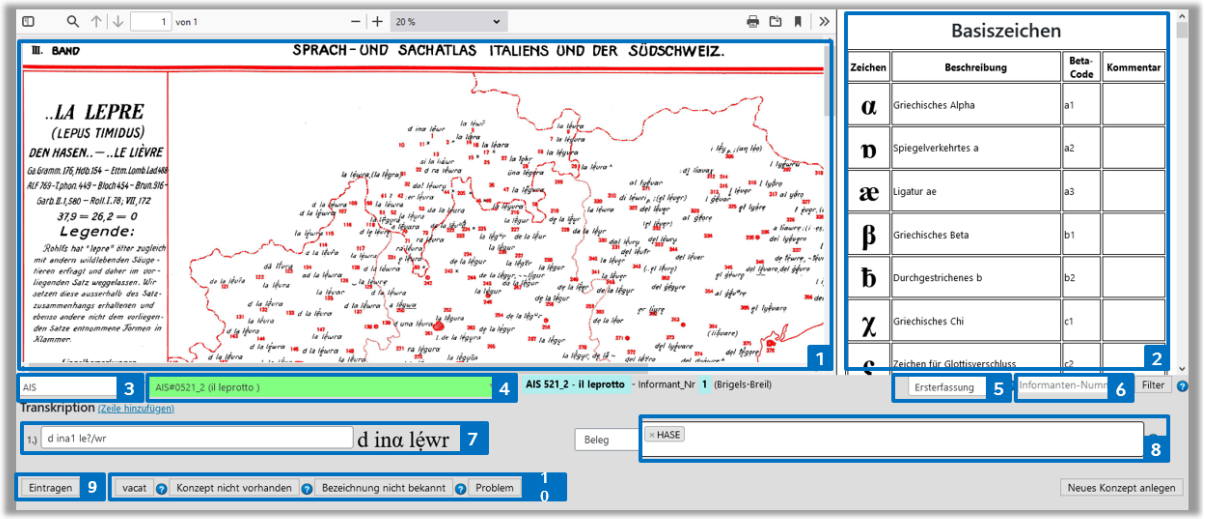
The differentiation between base signs (letters) located at the baseline and diacritics marked above and below base signs is crucial. Special sizes or positions of letters, for example when they are displayed smaller than other base signs in the source, are treated like diacritics.

Base signs that exist as ASCII characters are retained: That is to say all Latin characters. They are transcribed by a combination of a letter and a numeral. Diacritics are always placed after the base sign to which they are assigned. When several diacritics are assigned to one base character, they are transcribed from the bottom left to the top right. Each ASCII character used for the transcription of a diacritic may occur only one time per base character. For the repetition of the same diacritic, the Beta Code requires the use of special rules, e.g. \2 for a double grave accent. For the complete list of transcription rules cf. Lücke and Zacherl (2020).

Figure 2 *Differentiation between base sign and diacritics and transcription order*

The transcription tool was designed within the project. The main screen that appears using the tool contains first of all an original map from the source. On the side, there are the transcription rules. Below the map are two filters that allow to select the source and the map (stimulus, i.e. the question asked by the explorer during the data collection) that has to be transcribed. Once the map has been selected, the system asks to enter transcriptions for each survey point. The transition from one point to another is automatic: for each source, the system recognises which points are geographically located within the Alpine Convention and offers only those points for the transcription. The window for entering the transcription using the Beta Code rules is located below the filters and, to its right, the space for selecting and applying a concept to the transcribed utterance.

Figure 3 *The transcription tool of VerbaAlpina*



1 – Original map
2 – Transcription rules
3 – Data source
4 – Stimulus
5 – Entry/correction/problem
6 – Selection of the informant number
7 – Transcription in Beta Code and original representations
8 – Concept assignment
9 – Entry button
10 – alternative utterance format

2.2 Tokenization

The tokenization represents the second step towards the unification of the linguistic material. Through tokenization, the linguistic expressions which were transcribed, imported or entered into the database via crowdsourcing are fragmented into single tokens. After tokenization, the linguistic expressions are prepared to be used for the third step of the data processing, i.e. for the typification.

Table 1 *Overview of the tokenization process*

Attestation in Beta Code	**Attestation in IPA**	**Concept**
una1 mu:g/a1 da1 va/c)/	unɑ myʤɑ dɑ vˈatɕ	HERD OF COWS
TOKENIZATION		
una1	unɑ	ARTICLE
mu:g/a1	myʤɑ	HERD
da1	dɑ	PREPOSITION
va/c)/	v ˈatɕ	COW

Tokenization is carried out through a special tool that was developed within the project by entering the ID-number of the stimulus which corresponds to the map that one wants to tokenize. At this stage, if transcription errors are detected or if it is not possible to convert a string into IPA, the system displays the notification so that any errors can be corrected before the linguistic material is tokenized. Once tokenized, the linguistic attestations can be found via the interactive map. The search works through different filters which allow searching data from an onomasiological perspective (from the concept to the designations) as well as in a semasiological way (from the designation to the concepts) (cf. chapter 3).

2.3 Typification

After they have been tokenized, linguistic attestations can be further typified. The morphological typification of the linguistic data, which means the grouping of data according to their internal linguistic characteristics, is one core task of VerbaAlpina. Through the typification, VerbaAlpina aims at structuring the complex variety of the numerous linguistic attestations (tokens) so that comparisons among data are possible. A morpho-lexical type is defined by the following properties: language family, part of speech, single word vs. affixed words, gender, lexical base type. The morpho-lexical types represent the central category in the management of linguistic data and they are comparable to the lemmas which are listed in dictionaries.

Morpho-lexical types are specific to one language family. The form by which a morpho-lexical type should be represented (also in the search function of the interactive map) is given by the lemmas of selected reference dictionaries such as DWB (for the Germanic varieties), TLFi and Treccani (for the Romanic varieties) and SSKJ (for Slavonic) (cf. Krefeld and Lücke 2020b). In the case of Germanic and Slavonic tokens, it is rather easy to find the appropriate form because these two language families are represented by only one standardised language (German and Slovenian). For example, all the phonetic forms of Alemannic and Bavarian, which are variants of one morpho-lexical type, can be grouped under the same standard form. If standard variants do not exist, the lemmas of the most important regional reference dictionaries (Idiotikon, WBOE) are used (cf. Krefeld and Lücke 2020b). In the case of the Romance language family, the situation is more complex because of the numerous minor languages, some of which are still not sufficiently standardised. VerbaAlpina decided to represent all morpho-lexical types by the French and Italian standard forms (e.g. *beurre/burro* ‘butter’; *lait/latte* ‘milk’). If only one of these two standard languages has a suitable form, the type is constituted by this single term as in the case of *ricotta*, for which any corresponding form lacks in French. Similar to Germanic and Slavonic forms, if neither TLF nor Treccani contains any appropriate lemma, VerbaAlpina uses dialectal reference dictionaries (LSI, BLad) (cf. Krefeld and Lücke 2020b).

The typification of the linguistic material represents a further step towards the harmonisation of data, but above all, it allows to highlight linguistic convergences and divergences between the three language families of the Alpine region.

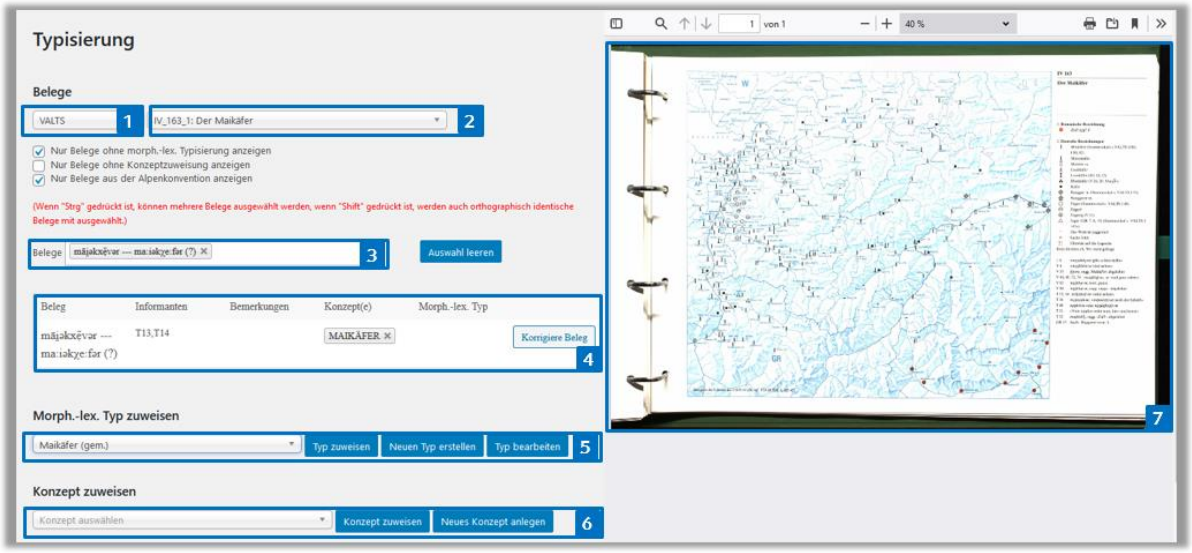
Each morpho-lexical type is described by entering linguistic information such as language family, part of speech, presence of affixes, and gender. At least one lemma from one of the reference dictionaries mentioned above is then applied to each morphological type. In addition, a so-called "base type" is applied to each morpho-lexical type. A base type is to be understood as the first historical attestation from which morpho-lexical types developed. In many cases, the base type corresponds to the etymon, with the only difference that an “etymon” refers to the immediate historical antecedent of a word, while a base type, as the name suggests, represents the older form (cf. Krefeld and Lücke 2018). Specific reference dictionaries (e.g. FEW, Georges, AWB) are also used for the selection of the base types.

Table 2 *Example of typification*

Token	kˈa:vra	kabrˈuŋ	kavrˈɛt	kawrˈɛt
Language family	roa	roa	roa	roa
Part of speech	noun	noun	noun	noun
Affix	-	+	+	+
Gender	f	m	m	m
Morpho-lexical type	capra	caprone	capretto	capretto
Base type	lat. capra	lat. capra	lat. capra	lat. capra

If a base type is marked as unknown or as not sufficiently clear by the reference dictionaries, VerbaAlpina applies a question mark as in the example *(?) battuere*. When it is not possible to determine the base type, we tend to use an unknown type represented by only one question mark.

Figure 4 *The typification tool of VerbaAlpina*



1 – Data source (linguistic atlas)
2 – Stimulus
3 – Selection of the attestations
4 – Attestations and properties
5 – Selection / creation / editing of morpho-lexical types
6 – Concept assignment
7 – Original map

3. Publication and Visualization

VerbaAlpina provides two main access points to the collected linguistic information, namely an interactive map and the so-called *Lexcion Alpinum*, a dictionary-like view on the data. This goes in accordance with the traditional ways of publishing in either linguistic atlases or dictionaries. A key difference to these established sources is the manner in which the information is provided. Whereas a linguistic atlas generally offers an onomasiological perspective and a dictionary a semasiological one with no (or at least very restricted) possibilities to approach the other way round, the digital counterparts that are developed for VerbaAlpina intentionally allow both perspectives each and therefore underline that the existing restrictions are mainly imposed by the paper-based publication form and not by the basic concepts of information representation.

3.1 Geolinguistic representation

As mentioned in chapter 2, all of the utilised linguistic data is geographically classified on principle which is a necessary condition to spatially visualize it in its entirety. To attain a presentation as consistent as possible VerbaAlpina utilises the respective administrative municipalities as a reference system, i.e. each linguistic attestation that is documented within the borders of a municipality is assigned to its geometrical centre.[6]

In addition to the linguistic core data, there are multiple other geo-referenced datasets added for the visualisation that are not strictly linguistic, but contain information relevant especially in the Alpine context. This includes areal data (e.g. the traditional language areas or administrative borders like nation-states, districts etc.) and point data (e.g. archaeological find spots or ancient and contemporary infrastructure placements). The principle idea in including this material is to allow the user to combine it with the linguistic data at will and thereby assist them to find patterns and draw conclusions about extra-linguistic factors that affect linguistic phenomena.

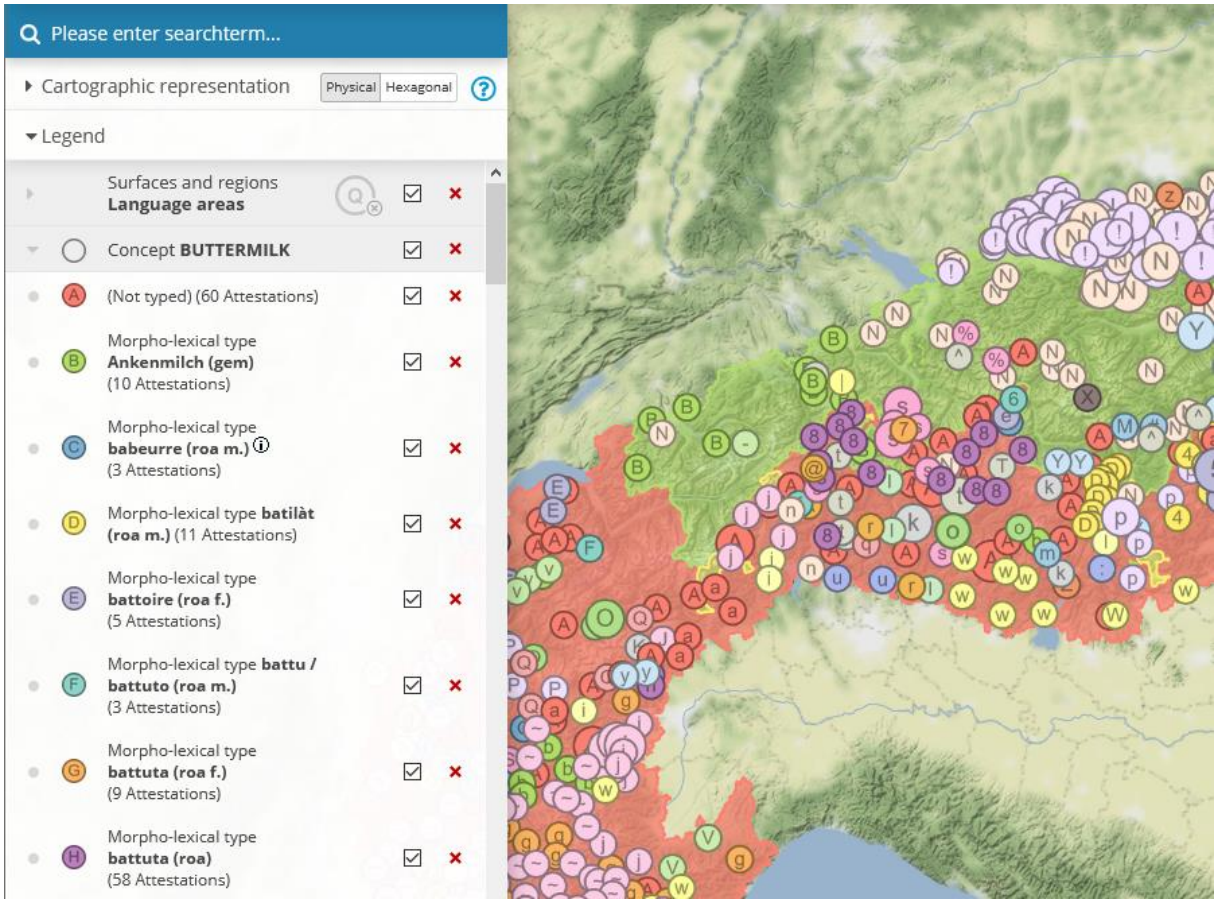
3.1.1 The Interactive Map

The main design choice concerning VerbaAlpina's interactive map is to allow the user to freely choose from the available linguistic and extra-linguistic data and to combine, filter and visualise in a way that fits their needs and interests best. Consequently, it starts off with an empty map that only contains the language areas in the Alpine region (which also can optionally be removed). The section *Cartographic representation* then allows to add new data to the map whereas the section *Legend* explains the symbols and colours that are shown on the map for the current selection.

The linguistic attestations themselves can be selected and grouped by three principal categories which occurred in the previous chapters: Morpho-lexical types, base types and concepts. Each one allows to show all attestations that are connected to the respective category and to group them appropriately. For example, the selection of a morpho-lexical type or base type allows (among others) grouping by concept which represents the classical semasiological perspective, whereas the selection of a concept vice versa allows grouping by one of the type variants to create an onomasiological representation. Figure 5 illustrates that for the concept BUTTERMILK and the respective morpho-lexical types.

[6] The interactive map allows this default behaviour to be changed via the options menu and each attestation to be shown exactly at the point where it is localised. This can be useful for source material with a particularly tight informant net in which many informants are located within one municipality.

Figure 5 *Legend and map representation for the selection of the concept* BUTTERMILK

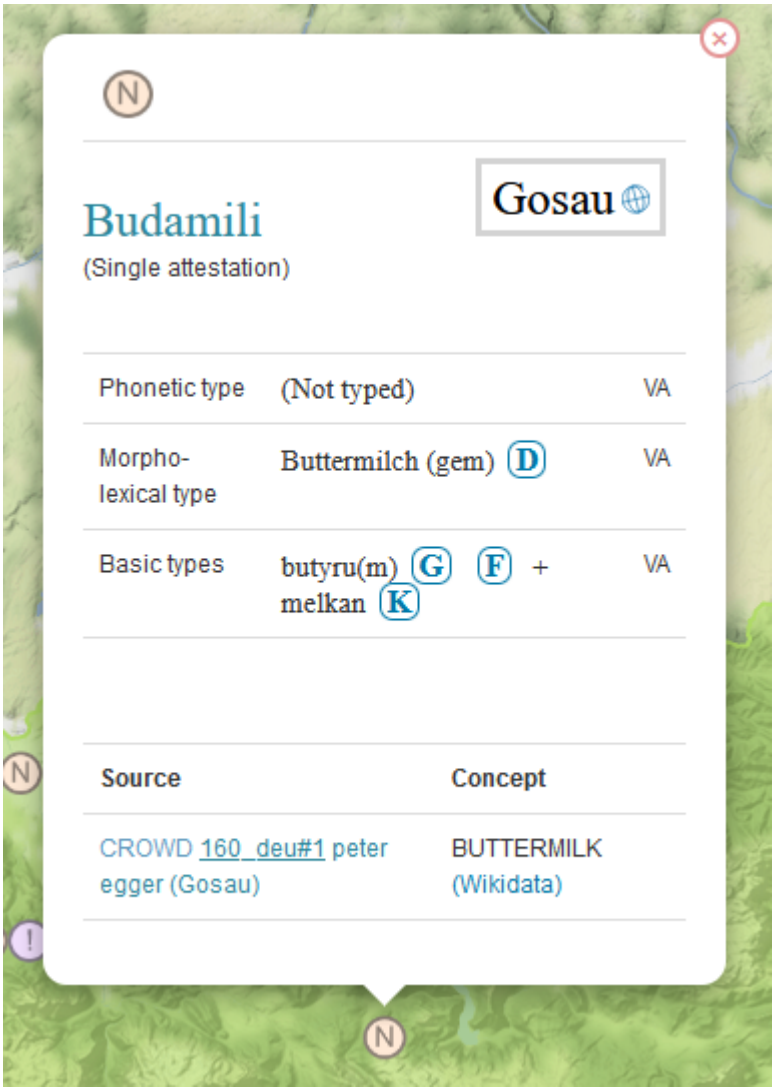


Depending on the particular category, there are also various filter and sorting possibilities to further customize the display. Additionally, the elements in the legend can be manually filtered so that sub-elements which are not relevant can be hidden or removed.

All point data (either linguistic attestations or extra-linguistic point data) is visualised on the map using coloured symbols that contain letters or numbers. The main category is specified by the symbol shape (circle, square, hexagon, etc.) whereas the colour and letter/number indicate the sub-category (cf. Figure 5). This allows multiple groups of attestations to be shown at once while they still can be distinguished from each other. Multiple symbols at the same location are joined to a larger symbol which slowly grows on a logarithmic scale until it reaches a maximum size. All areal data is represented by coloured part-transparent polygon overlays.

While the map representation alone gives an overview over the distribution of the different types or concepts in space, it is also possible to access the full information about one single attestation. Following the so-called *visual information seeking mantra* "overview first, zoom and filter, then details on demand" (Shneiderman 1996) this can be achieved by opening an extra popup window for each overlay on the map. Among others this contains details about the utterance itself, its typification and the source it came from. Many elements in this window are interactive to give further information or link to external resources like the dictionary references added during the typification process.

Figure 6 *Detail view for a specific attestation*

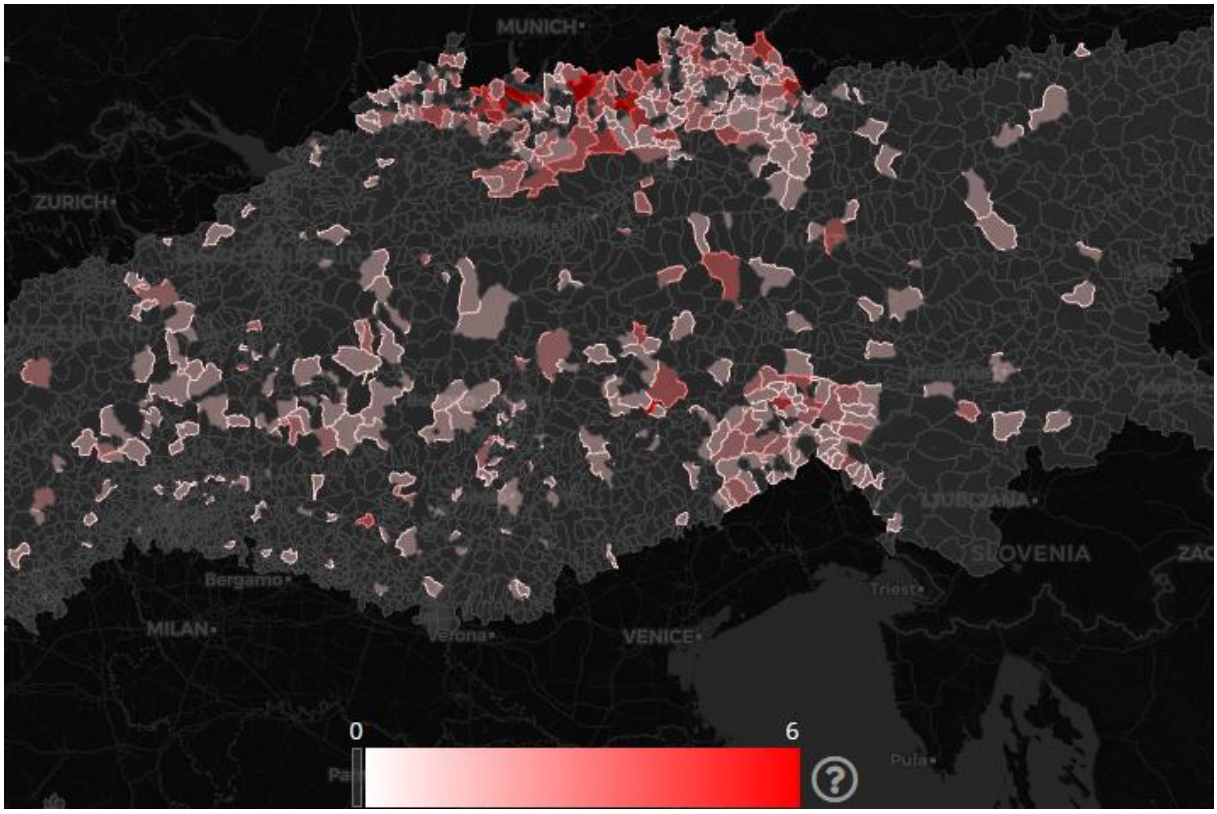


An important feature of the interactive map is the possibility to store its current state. This includes data selection, position and zoom of the map and other options like the different visualisation modes presented in the next chapters. In principle there are two possibilities to achieve this: The creation of a so-called *synoptic map* or the creation of a share-link. Technologically both variants store the current state in the VerbaAlpina database, but while a synoptic map contains a name and a description and can be made available for other users, the share-link solely produces an URL which can be used to re-create the current state.

3.1.2 Qualitative vs. Quantitative Presentation

In addition to the symbol-based visualisation that was presented in the previous chapter and which is activated by default, the interactive map offers a second, more aggregated way to show the data. The *quantitative view* produces a heat map to depict the distribution of attestations. It is possible to choose any of the polygon layers provided and project the current data selection on it using the *Q* button in the respective legend entry. Figure 7 shows the map from figure 5 in quantitative view.

Figure 7 *Quantitative view for the concept* BUTTERMILK

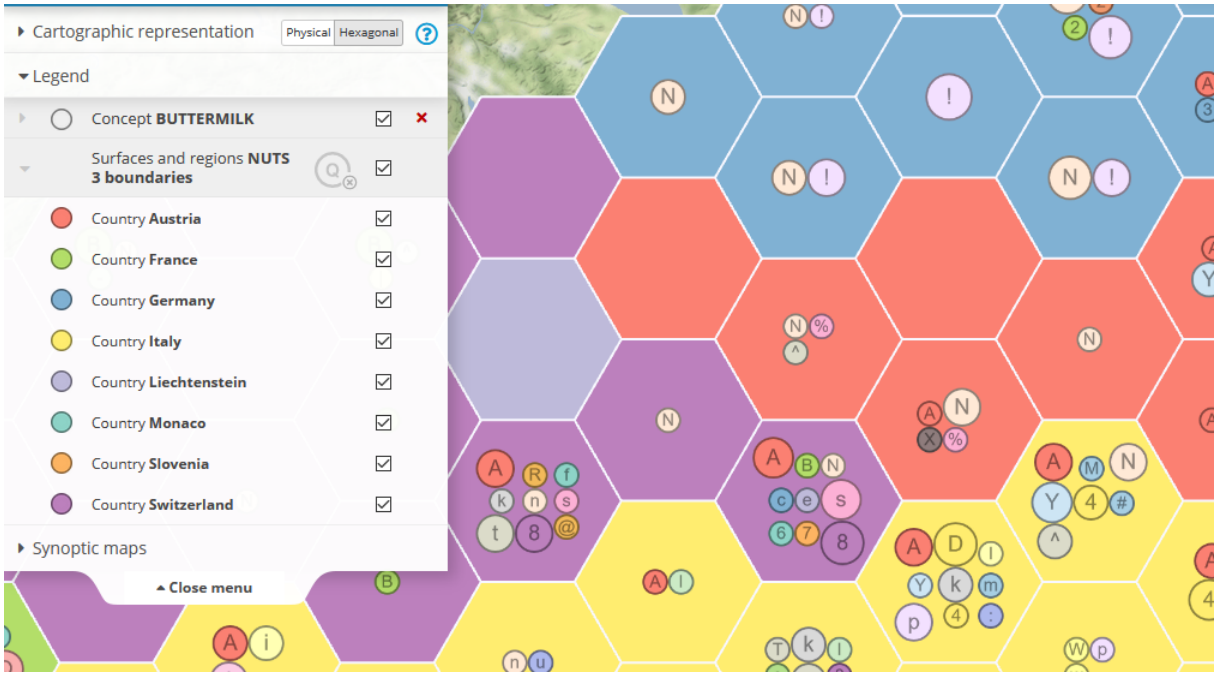


While the normal qualitative representation allows to find patterns in the visualised data and to access all details, the quantitative view can be used for more specific tasks. It allows the evaluation of the denseness for a certain portion of the data, which on the one hand helps internally to find gaps in the data, which for example can be specifically addressed by promoting crowdsourcing for certain areas. On the other hand, it is also useful for interpretation of the visualised data as it can be compared with population density maps or the like and may also provide information about the geographic focus of the data material from specific sources.

3.1.3 Geographic vs. Abstract Presentation

A second possibility to alter the nature in which the interactive map is displayed is to change to the abstract mode. In opposition to the *geographic mode* the different polygon layers are simplified to hexagons and arranged in a grid that tries to retain the neighbourhoods of the original polygons as close as possible. At the moment this is only possible for the so-called NUTS-3 borders, that are administrative borders one level above the municipalities, and the languages areas. In the future another grid consisting of the municipality borders will be added. In qualitative mode the point symbols are arranged at the centres of the respective polygons, in quantitative mode the hexagons are coloured just like the original polygons.

Figure 8 *Abstract view for the concept* BUTTERMILK

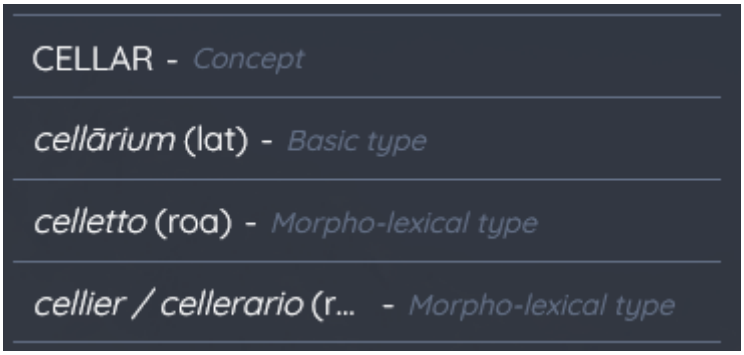


The main reason for introducing this extra mode of visualisation is to counter the effect that larger regions might be considered more important than smaller regions. Especially in the Alpine region, large municipalities or other administrative units are often very thinly populated whereas bigger towns show up very small on a geographic map. If each one of them is presented with the same size, this effect is removed, although precision and the direct recognition of specific geographic features get lost.

3.2 Lexicographical representation

The second entry point to the collected VerbaAlpina data is the dictionary-like *Lexicon Alpinum*. Its surface consists of a title bar on the left and the main area that contains a list of open articles. The title bar lists all elements from the main categories (morpho-lexical types, base types and concepts) in alphabetical order. Figure 9 shows a small section that illustrates this.

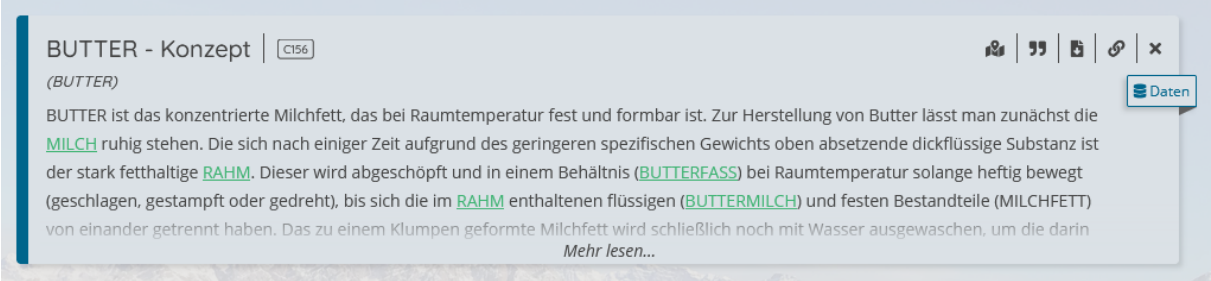
Figure 9 *Excerpt from the article list of the Lexicon Alpinum*



Each entry is visualised on a card which has a front and a back side. The front side contains general information about the specific entry and links to connected sites like the representation on the map and an API call to receive the underlying data. If existing, it also contains a comment explaining specific

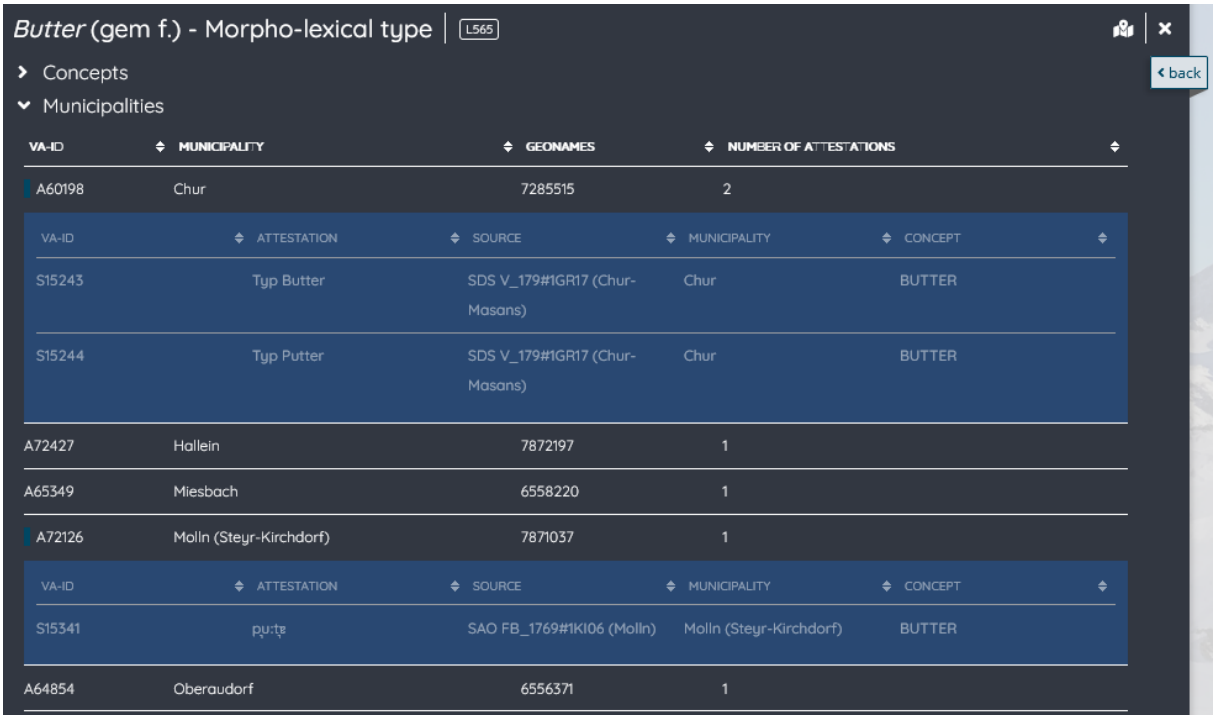
information about this entry[7] and a template to cite this specific entry. The majority of entries does not have an explanatory text, though, since it is only added, if there are no other online resources available which contain this information. Especially for concepts the connection to Wikidata (and indirectly to Wikipedia) generally holds more value than an own mostly redundant description.

Figure 10 *Front*



The back side allows to access the actual linguistic data in detail. Like for the map, there are different sub-categories depending on the main category, e.g. concepts for both kinds of types and types for concepts vice-versa. Like mentioned before, this also mirrors the two traditional lexicographic perspectives on the data. Also, all entries contain a sub-category *Municipalities* which allows to group all attestations geographically (cf. figure 11). Hence it is possible to get a representation very similar to a traditional dialectal atlas that lists variants for a number of locations. For every sub-category the elements can be further extended to show information about the specific attestations, similar to the details info window on the interactive map.

Figure 11 *Back*



[7] This text can also be accessed via the legend of the interactive map.

Analogously to the feature of the interactive maps, it is similarly possible to store the current selection in the *Lexicon Alpinum* by creating a share link. This stores the opened articles as well as more specific details (the position which the user scrolled to, if the front or back side of an article is shown, which elements are extended, etc.)

4. Conclusion

This paper describes the major tools that are used in the VerbaAlpina project to process and publish linguistic data. All program code for these tools is made public in half-year intervals via multiple GitHub repositories and can be found here: https://github.com/VerbaAlpina/. In addition to the main tools, which mostly are realized as separate WordPress[8] plugins, there is a multitude of smaller tools that are exclusively used internally for specific tasks. Examples are a small tool which allows the editing of the hierarchical representation of concepts and sub-concepts or one that checks the text published on the VerbaAlpina homepage for structural errors. These are (like the main tools) developed in close collaboration between the linguistic and informatics staff and are probably too specific to be used in different contexts. However, they are included in the main VerbaAlpina plugin repository at the previously given URL.

VerbaAlpina also provides an API through which the data can be accessed in defined formats programmatically. The selection of the data and the output format are controlled by URL parameters. The API of VerbaAlpina can be found at https://www.verba-alpina.gwi.uni-muenchen.de/?page_id=8844&db=xxx.

References

AWB = Karg-Gasterstädt, Elisabeth and Frings, Theodor (eds.) 1952-2015: *Althochdeutsches Wörterbuch, digitalisierte Fassung im Wörterbuchnetz des Trier Center for Digital Humanities*, http://awb.saw-leipzig.de/ (accessed 27.05.2021).

BLad = Istitut Cultural Ladin Majon Di Fascegn 2007: *Banca lessicala ladina*. Vigo di Fassa, http://blad.ladintal.it/applications/dictionary/siteHistoric/index.jsp?_VP_V_ID=834120465 (accessed 27.05.2020).

Colcuc, Beatrice and Krefeld, Thomas 2020: *Sprachatlanten und Wörterbücher im Alpenraum*, Methodologie, VerbaAlpina-de, https://doi.org/10.5282/verba-alpina?urlappend=%3Fpage_id%3D493%26db%3D202%26letter%3DS%2352.

DRG = De Planta, Robert, Melcher, Florian, Pult, Chasper, and Giger, Felix 1938–1940: *Dicziunari Rumantsch Grischun*. Chur: Institut dal Dicziunari Rumantsch Grischun, http://online.drg.ch/ (accessed 27.05.2020).

DWB = Grimm, Jacob and Grimm, Wilhelm 1854-1961: *Deutsches Wörterbuch von Jacob und Wilhelm Grimm, digitalisierte Fassung im Wörterbuchnetz des Trier Center for Digital Humanities*, Version 01/21, https://woerterbuchnetz.de/?DWB, (accessed 27.05.2020).

FEW = Wartburg, Walter 1922-1967: Französisches etymologisches Wörterbuch. Basel: Zbinden, https://apps.atilf.fr/lecteurFEW/ (accessed 27.05.2020).

[8] https://wordpress.org/.

Georges = Georges, Heinrich 1913–1918: *Ausführliches lateinisch-deutsches Handwörterbuch*. Aus den Quellen zusammengetragen und mit besonderer Bezugnahme auf Synonymik und Antiquitäten unter Berücksichtigung der besten Hilfsmittel ausgearbeitet. Unveränderter Nachdruck der achten verbesserten und vermehrten Auflage. 2 Bände. Darmstadt: Wissenschaftliche Buchgesellschaft, 1998 (Reprint). Hannover: Hahnsche Buchhandlung, http://www.zeno.org/georges-1913 (accessed 17.05.2020).

Gilliéron, Jules and Edmont, Edmond 1902–1910: *Atlas linguistique de la France*. Paris: Champion.

Idiotikon = Schweizerisches Idiotikon. *Schweizerdeutsches Wörterbuch*. Basel. http://www.idiotikon.ch/index.php (accessed 26.05.2020).

Jaberg, Karl and Jud, Jakob 1928–1940: *Sprach- und Sachatlas Italiens und der Südschweiz*. Ringier: Zofingen.

Krefeld, Thomas and Lücke, Stephan (eds.) 2014: *VerbaAlpina*. Der alpine Kulturraum im Spiegel seiner Mehrsprachigkeit. München: online, https://dx.doi.org/10.5282/verba-alpina.

Krefeld, Thomas and Lücke, Stephan 2018: *Typisierung, Methodologie*, VerbaAlpina-de 20/2, https://doi.org/10.5282/verba-alpina?urlappend=%3Fpage_id%3D493%26db%3D202%26letter%3DT%2358.

Krefeld, Thomas and Lücke, Stephan 2020a: *Crowdsourcing, Methodologie*, VerbaAlpina-de 20/, https://doi.org/10.5282/verba-alpina?urlappend=%3Fpage_id%3D493%26db%3D202%26letter%3DC%2312.

Krefeld, Thomas and Lücke, Stephan 2020b: *Referenzwörterbücher, Methodologie*, VerbaAlpina-de 20/2, https://doi.org/10.5282/verba-alpina?urlappend=%3Fpage_id%3D493%26db%3D202%26letter%3DR%2351.

Krefeld, Thomas 2020: *Sprachen und Sprachfamilien im Alpenraum, Methodologie*, VerbaAlpina-de 20/2, https://doi.org/10.5282/verba-alpina?urlappend=%3Fpage_id%3D493%26db%3D202%26letter%3DS%2353.

LSI = Lurà, Franco (ed.) 2004: *Lessico dialettale della Svizzera italana*. Bellinzona: Centro di dialettologia e di etnografia.

Lücke, Stephan and Zacherl, Florian 2020: *Transkriptionsregeln, Methodologie*, VerbaAlpina-de 20/2, https://doi.org/10.5282/verba-alpina?urlappend=%3Fpage_id%3D493%26db%3D202%26letter%3DT%23150.

Shneiderman, Ben 1996: *The eyes have it: a task by data type taxonomy for information visualizations*. Proceedings 1996 IEEE Symposium on Visual Languages, 336–343.

SSKJ = *Inštitut za slovenski jezik Frana Ramovša: Slovar slovenskega knjižnega jezika*, http://www.fran.si/130/sskj-slovar-slovenskega-knjiznega-jezika (accessed 28.05.2021).

TLFi = *Trésor de la langue française informatisé*, http://atilf.atilf.fr/tlf.htm (accessed 19/05/2021).

TLG = Brunner, Theodore (ed.) 1972-, *Thesaurus Linguae Graecae*. Irvine: University of California, http://stephanus.tlg.uci.edu/ (accessed 24/05/2021).

Treccani = *Vocabolario Treccani online*, https://www.treccani.it/vocabolario/ (accessed 19/05/2021).

WBOE = Bauer, Werner, Kranzmayer, Eberhard and Institut für österreichische Dialekt- und Namenlexika (eds.) 1970: *Wörterbuch der bairischen Mundarten in Österreich*. Wien: Verlag der Österreichischen Akademie der Wissenschaften.

Wilkinson, Mark and Dumontier, Michel et al. 2016: *The FAIR Guiding Principles for scientific data management and stewardship*. Scientific Data 3, https://www.nature.com/articles/sdata201618 (accessed 25.05.2021).

DOI: https://doi.org/10.14232/wpcl.2022.7.4

Not sustainable but beautiful? – Some steps towards visual access to multidimensional data collections

Timm Lehmberg

1. Introduction

The following report provides insights in the data analysis and visualization practice of the long term language documentation project INEL.

The principles developed try to take into account the demands of data curation in this special type of projects that lie in the field of tension between sustainability and long term preservation on the one side and short term needs of visual access to the resources on the other side.

A central aspect is to rely on widely adopted and in some cases even proprietary tools and platforms that reduce the effort of interface building and generate maximum flexibility and scalability.

2. Initial Position and Requirements

The 18-year longterm project INEL (Grammar, Corpora, Language Technologie for Indigenous Northern Eurasian Languages) aims at the curation and analysis of language data coming from endangered languages/varieties of the Northern Eurasian Area[1].

Having started in 2016 the project generates deeply annotated digital language corpora and further resources which are made long term available both to the scientific and speaker communities as well as the interested public.

For this purpose, INEL is structured into language specific three-year sub-projects that, following predefined workflows, deal with the curation (in some cases new acquisition), time-aligned transcription, glossing and multi-layer annotation of language data in the respective language/variety. In this way up to the present-day corpora in Selkup, Dolgan, Kamas, and Evenki language have been finalized and published under open access conditions. (A more comprehensive description of the project aims can be found at Arkhipov and Däbritz 2018.)

However, beyond long term availability and sustainability the project considers its mission to provide visual interfaces that comply both to the demands of various target audiences and project research which of course puts high demands on its research data management.

While for example proven solutions to sustainability issues beyond other things result in the use of rather static storage types using established and in many cases generic data standards and formats (i.

[1] The project is funded by 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.

Nicole Palliwoda (ed.): *Data Processing and Visualization in Variational Linguistics/Dialectology*. Working Papers in Corpus Linguistics and Digital Technologies: Analyses and methodology- Vol.7. 2022.

e. markup standards as defined by Text Encoding Initiative TEI[2]), the creation of visual interfaces[3] follows rather short term dynamics. This might be caused both by the rapid evolution in the area of visualization technology and also by the fact that approaches on data visualization and GUI creation often have to react to short term demands like scientific analyses or data search requirements. Furthermore, it is to mention that GUI creation requires a high level of personal effort which usually cannot be provided by time-limited and in most cases third-party funded research projects.

In order to reduce the effort of interface development and to stay flexible with respect to upcoming analyses and visualization-demands, a multi-level approach was chosen that on its ground level makes use of sustainable data structures and uses flexible tools for indexing and data analyses on top that allow for flexible and user friendly GUI creation.

In the following first the tools that are used for data processing and, based on this, the resources that are created and will be introduced. Based on these concrete examples for resource overarching visualization approaches and analysis will be presented.

3. Tools, platforms and data standards

EXMARaLDA

Depending on the nature of the primary data, a variety of tools like i. e. FLEx and ELAN are being used for the pre-processing of the language data (esp. transcription and glossing). However, finalization, analysis and publication is done exclusively with the help of EXMARaLDA4 (Schmidt and Wörner, 2014), a widely established framework of platform independent desktop applications and data formats to be used along with spoken language corpora.

The major reason for using EXMARaLDA is, besides its wide range of features and tools, the consequent use of XML-based and thus sustainable and interoperable data formats.

An important recent development that has to be mentioned in this context is the establishment of the e. ISO/TEI standard “Transcription of Spoken Language” on the base of the EXMARaLDA data formats.

Corpus Services

Based on the findings from longtime curation work at spoken language corpora the need for a modular and scalable tool collection arose, that would allow for an easification and, if possible, automation of recurring data checks and fixes. As a result Corpus Services, a collection of java based tools was developed initially at the Hamburg Centre for Language Corpora (HZSK) and in the following utilized and further developed in the projects CLARIAH-DE[5], CLARIN-D[6], INEL and QUEST[7]. It contains functionality used for data maintenance, curation, conversion, and visualization, primarily with a focus on data formats used in the environment of EXMARaLDA based spoken language corpora (see above).

[2] https://tei-c.org/

[3] In the following the acronym GUI for graphical user interfaces will be used for any kind of visual interface.

[4] https://exmaralda.org

[5] https://www.clariah.de/

[6] https://www.clarin-d.net/

[7] https://www.slm.uni-hamburg.de/ifuu/forschung/forschungsprojekte/quest.html

In the INEL project with the help of corpus services nightly automated checks and fixes were implemented that produce extensive logging information on errors or weak points in the data and thus found a solid base for a high quality of the research data already in the process of creation.

Tsakorpus

The Tsakonian Corpus platform[8], is a corpus search platform that provides a web based search interface which not only allows intuitive multilayer search over annotations and glosses but also provides audio-aligned search results (if existing). The Tsacorpus backend is implemented with the help of *elasticsearch* and thus enables data querying that combines flexibility with respect to data models and schema with a high level of indexing and thus querying performance (see below). These major characteristics make Tsakorpus a powerful tool for the web-based corpus search in the project. An important step towards the seamless integration of INEL corpora into the Tsakorpos platform on has been done by Arkhangelskiy et al (2019) who define a workflow for the integration of audio-aligned transcripts formatted in the ISO/TEI standard for "Transcription of Spoken Language" mentioned above.

Elastic Stack

The Elastic Stack is a collection of open source software tools used for the analysis of textual data that is developed and distributed by the company elastic[9]. Its key component, the search engine elastic search (which is based on Apache Lucene[10]), is considered to be one of the most frequently used search engines in production environments. Further components used along with data analysis and visualization in the INEL project are *Logstash* (used for defining and executing ingest pipelines into elasticsearch) and *Kibana,* which provides comprehensive web based and interactive visualization and analysis capability that can easily be setup and scaled with respect to individual research issues. The most common fields of application in digital production environments for the elastic stack are customized search engines, analyses of large amounts of (often time based) data like contained log files and complex platform overarching data visualization. In light of the considerable effort of the development of graphic user interfaces (Lehmberg 2020) Kibana becomes a powerful tool for the out-of-the-box creation of intuitive and user centered interfaces.

4. Core and Accompanying Resources

The spoken language corpora described above form the core of the INEL resources. The use of the EXMARALDA System allows both for sustainable long term availability and the search capabilities provided by the EXMARaLDA System.

However, it is not surprising that in the framework of an 18 year long term initiative that focuses on data acquisition and curation at every point in time additional linguistic resources (like catalogue, lexical and geospatial data etc.) appear to be crucial for research.

Their necessary integration into the project resources or at least the correlation of the information they contain with the project data put high demands on data formats and workflows and, with respect to analysis and visualization issues, require a performant and flexible set of tools to be available.

[8] https://github.com/timarkh/tsakorpus

[9] https://https://www.elastic.co/

[10] https://lucene.apache.org/

In the following a brief overview on these accompanying resources followed by a selection of visualization approaches will be given.

Bibliographic and catalogue data

More than in many other areas of linguistic research, the documentation and analysis of minority languages not only relies on the analysis of the object language data itself but also on information gained from secondary sources of information. The variety of these resources is considered to range from unstructured (often analogue) language data collections to lexicon data, catalogue data and personal notes created by researchers in the past (first and foremost *manuscript fieldnotes*, cp. Sanjek 1990, Sanjek/Trattner 2016). Of course, an essential role is also played by references to secondary literature which, due to the specificity and low documentation of the languages in focus, can be hard to find in published form.

As a reaction to this in the INEL project comprehensive catalogue resources are being created that contain information on secondary resources. These linear and semistructured resources are both used for internal purposes and, in cases where their coverage, structuredness and well-formedness achieve a certain level, made available to the public.

Two prominent resources that have to be mentioned in this context are to mention in this context are the INEL research Bibliography[11] and the digital edition of the Kuzmina archive (Lehmberg 2020).

Geodata

Modeling, visualizing and correlating spatial data puts high demands on the data workflows to be chosen in minority language documentation and analysis. Central problems often arise from the vagueness and ambiguity of the data, variation in the naming of geographic entities and also the fact that spatial information may occur on all layers of a language resource.

As an example of the latter spatial information with different granularity may occur in metadata (both for exploration settings and speaker biography), in the object language itself and also in the form of external resources.

Considering this high level of variety a structured and standardized modeling of all types spatial seems to be problematic (and maybe even not desirable). At the same time the variety and richness of data contains a lot of explicit or implicit information which are relevant for sociolinguistics, typology and many other research areas. As an example, the use of certain toponymes for the description of geo entities may provide information on the presence of language communities in a particular area and time, migration processes, language contact and many more.

5. Visualization Issues

The creation of visual and in an ideal case interactive graphical user interfaces for querying and analysing language data usually requires a high amount of personal and technical effort. It becomes even more complex if the data to be analyzed is structured and formatted rather with a focus on sustainability and elaborated data structure than on efficient querying like in the case of several well established markup based data formats used along with linguistic data. As a common approach, various

[11] http://doi.org/10.25592/uhhfdm.731

steps of transformation and linearization are to be made in order to generate user-friendly output which complies with the respective research issues.

For this reason, the INEL project follows the principle of a maximum adaptation of existing technology to be used for online-publication and visual (search-)interfaces. As an example, the INEL research Bibliography (see above) is stored and made long term available with the help of the BibTeX standard nut published for online-search as static HTML output with a Javascript based search interface, an output that can be created easily using the open-sourced reference manager tool *JabRef*[12].

As a further system that has been adapted in order to be used for out-of-the-box search, analysis and visualization the *Elastic Stack* (see section 2) was chosen. The following sections will provide a selection of insights into several application scenarios.

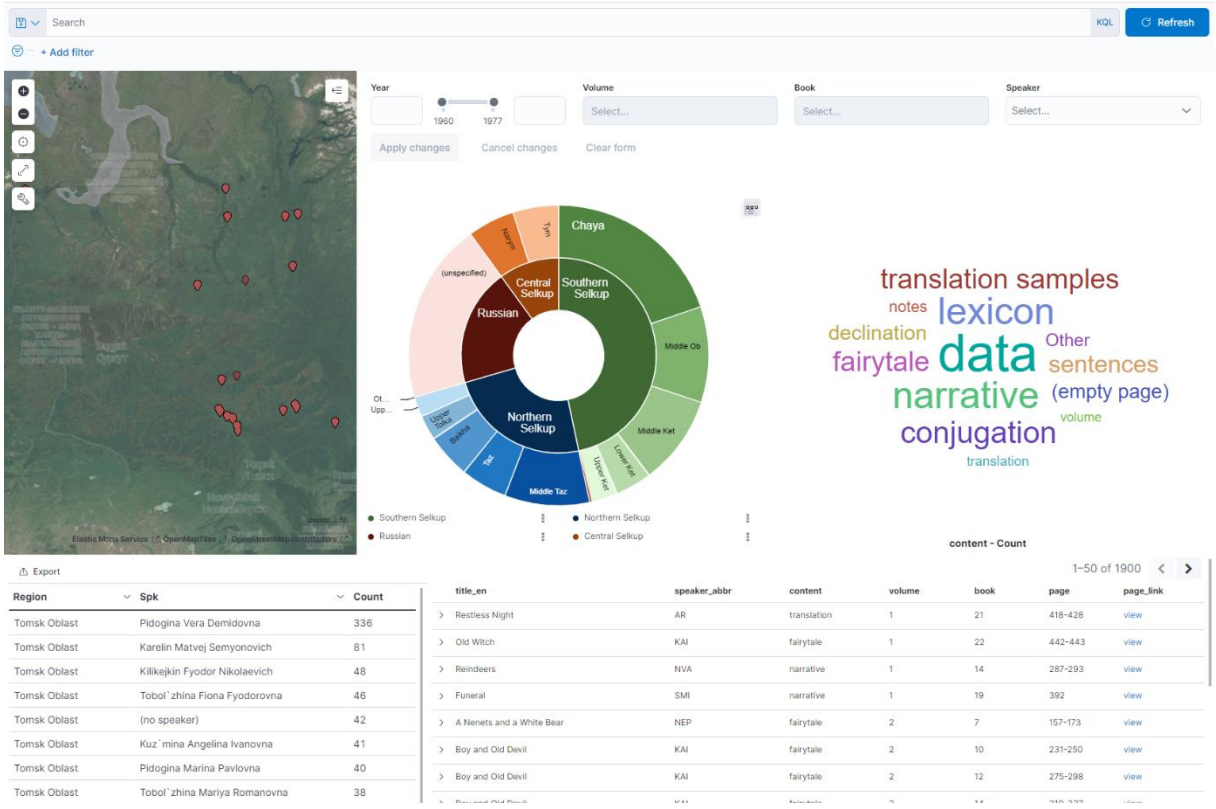
5 .1 Cross resource visualization and analysis

In the case of the abovementioned digital edition of the Kuzmina Archive that consists of digitized (scanned) manuscript pages and a well-structured and consistent tabular catalog (cp. Lehmberg 2020) for the first time in the INEL project a two-layered approach was chosen. To ensure long term availability the catalog is stored in TEI P5 compliant XML format along with scanned facsimiles in the research data repository of Universität Hamburg[13]. However, for analysis issues the catalog was ingested into the project's elastic cluster which then not only allows for effortless GUI creation with respect to the project's individual research demands but also straightforward correlation with other project resources (see Figure 1).

[12] https://www.jabref.org/

[13] https://www.fdr.uni-hamburg.de/

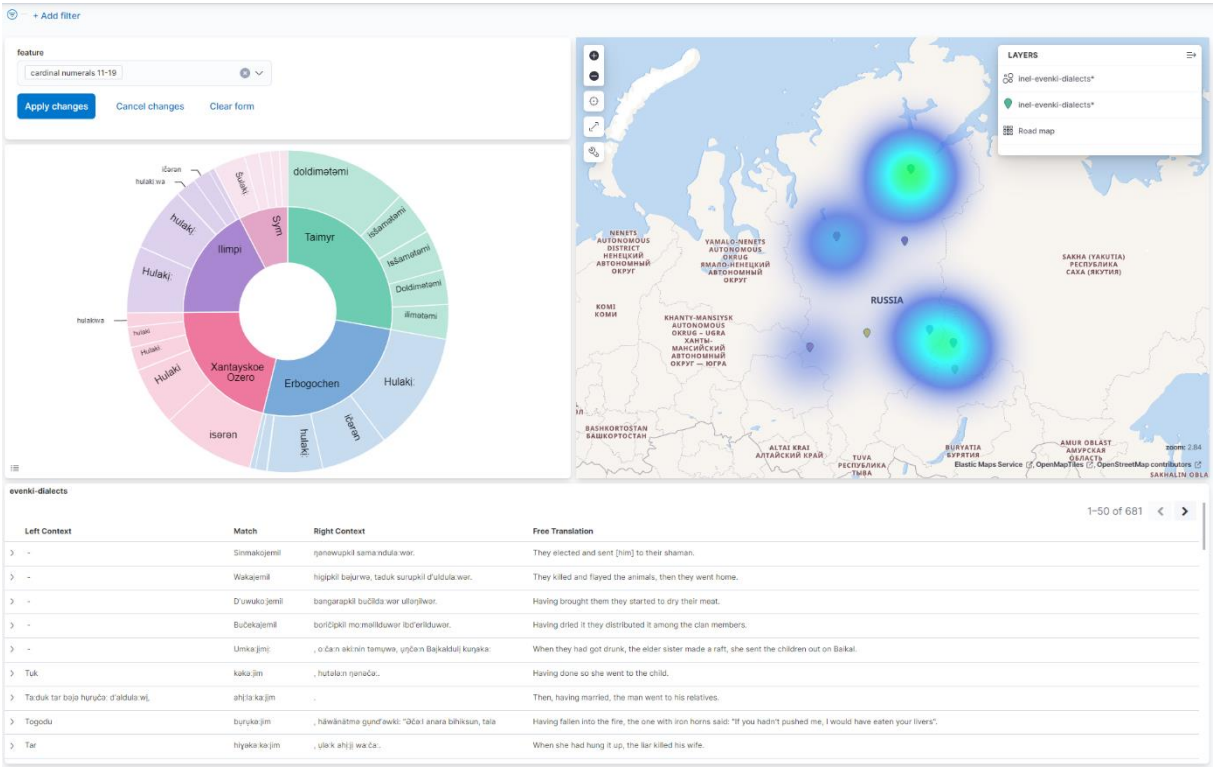
Figure 1 *Dashboard for straightforward access to the catalogized manuscript data of the 'Kuzmina Archive'*



After successful implementation further approaches on ingesting linear structured data, primarily derived from the INEL corpora, were taken. An essential role in this context was played by the tool EXAKT (EXMARaDLA Analyse und Konkordanzprogramm) a standalone tool which provides multilayer search using regular expressions over corpora stored in the EXMARaLDA format. As one central output format it provides linear structured concordances, possibly containing an arbitrary number of additional columns which for instance may contain annotation or metadata values associated with the respective match. These concordances that usually carry information corresponding to specific research issues found a perfect base to be ingested into a data analysis framework like the above mentioned elastic stack. In doing this, they can be correlated with other data to get new insights into the corpora with a minimum of conversion effort.

While earlier approaches of resource overarching analysis in the INEL project aimed at the definition of "linking" data (cp Jettjka/Lehmberg 2020) this method turned out to be much more flexible and applicable because it allows it to react to visualization demands already in the framework of the process of corpus creation. The following figure shows an exemplary visualization generated on the base of the INEL Evenki Corpus.

Figure 2 *Dashboard visualizing the distribution of dialect features of Evenki*



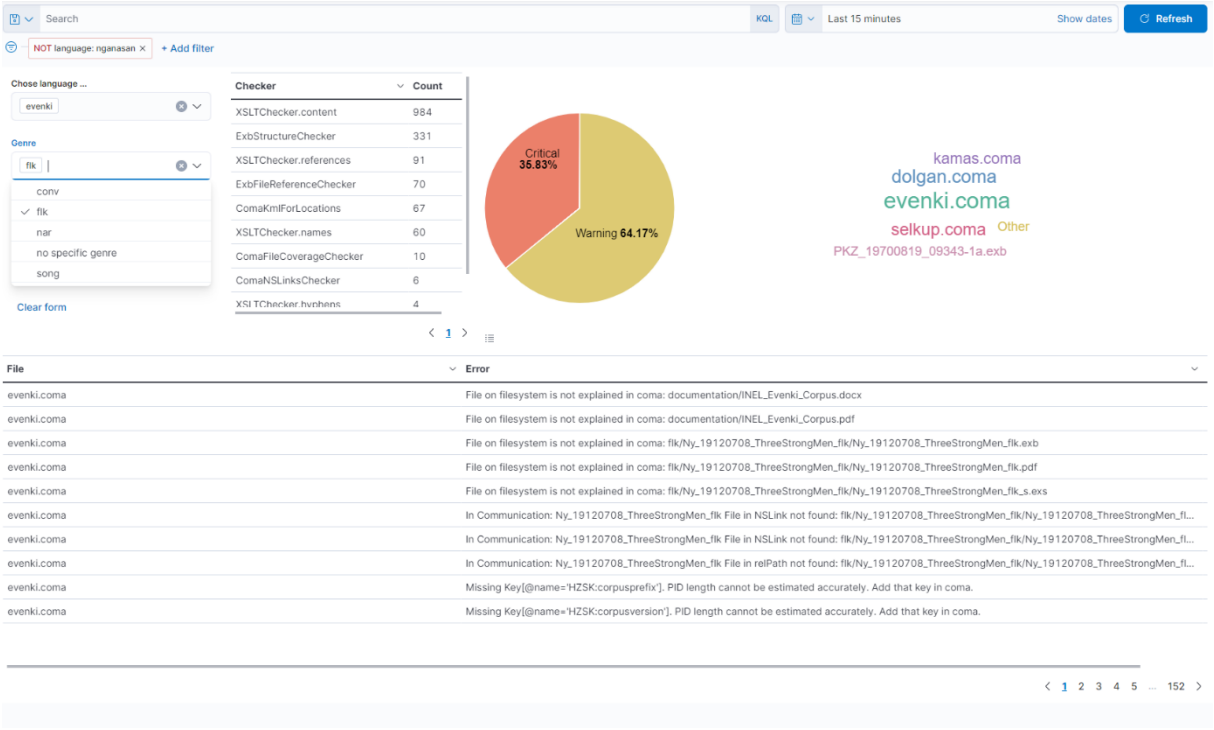
5.2 Visualization for automatic Quality Control and Consistency Checking

As mentioned above, to allow for already using the corpora and resources within the process of their creation, detailed workflows for all necessary steps (including, primary data curation, annotation and glossing, for a detailed description cp. Arkhipov/Daebritz 2018) where defined[14].

Furthermore, deriving the principles of *continuous integration* from software development projects (like described by Ferger/Hedeland 2020), daily consistency checks and (in many cases automatic) fixes are applied to the data. A central function in this process is taken over by the abovementioned software framework *Corpus Service* (see section 2). The output of these daily checks is stored in the form of comprehensive error lists containing information on inconsistencies in metadata, data structure, noncompliance of project specific conventions and further categories. In an initial step a hard coded html output and in addition XML based error lists that easily can be imported into the EXMARaLDA Partitureditor for error correction was chosen as output format. Facing the need for filtering, searching and a more intuitive access to the log files that results from the immense amount (in some rare cases more than 1000) of entries and also their redundancy, it was decided to index and visualize them with the help of the elastic stack. As a result, not only corpus overarching error search and filtering capability was added, also error prone files are highlighted automatically which led to an immense improvement of the user experience (see Figure 3).

[14] Workflowmanagement and -monitoring are proceeded with the help of an Issue tracing System

Figure 3 *Graphical Overview, results of INEL consistency checks*



5.3 Visualization for manual consistency checking

Both core and accompanying project resources contain information that is connected to entries in other project resources or appear in different resources on different layers. As an obvious example, named entities that carry geographic information occur in corpus metadata (for speakers and sessions), catalog resources and also in speaker utterances themselves. In order to be used for georeferencing and geovisualization and further resource overarching analyses they need to be harmonized with respect to writing (i. e. cyrillic writing and its latin transliteration) and the geo coordinates they refer to.

The same applies for the naming of individuals and bibliographic references. In cases where the necessary harmonization of these entries cannot be done by automatic consistency checks and fixes, indexing and creating aggregation based visualization allows for straightforward manual consistency checking by project members.

6. Conclusion and outlook

As shown in this report it seems to be a suitable approach to meet the visualization needs and requirements in scenarios like given in the INEL project by complying to established and widely adopted tools and standards on the one hand and making use of existing (often proprietary) out-of-the-box solutions for data analysis and visualization on the other hand.

In doing so it becomes possible to build the bridge between workflows that take into account the sustainability and long term availability of the data (which in many cases does not include user friendly visual access) and the everyday visualization demands resulting from for example from consistency checking and analysis. Further steps to be performed in the INEL project will be the correlation of indexed Tsakorpus search data (already stored in the form of elasticsearch indexes) and further (i. e. lexical) data.

However, though the principles and practices described here have been proven successful in short term visualization they have a number of significant disadvantages with respect to long-term

availability. By their very nature, visual interfaces created with the help of proprietary software frameworks like the elastic stack require continuous maintenance and sometimes also financial effort. This becomes even more crucial in cases where visualizations being referenced in publications (ideally using persistent identifiers or at least permalinks) have to be kept long term available.

References

Arkhangelskiy, Timofey, Ferger, Anne and Hedeland, Hanna: Uralic multimedia corpora 2019: ISO/TEI corpus data in the project INEL: *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, 115–124. https://aclanthology.org/W19-0310.pdf.

Arkhipov, Alexander and Däbritz, Chris Lasse 2018: Hamburg corpora for indigenous Northern Eurasian languages. *Tomsk Journal of Linguistics and Anthropology* (3): 9–18. https://doi.org/10.23951/2307-6119-2018-3-9-18.

Ferger, Anne, Hedeland, Hanna, Jettka, Daniel, and Pirinen, Tommi 2020: *Corpus Services* (Version 1.0). Zenodo. http://doi.org/10.5281/zenodo.4725655.

Hedeland, Hanna and Ferger, Anne 2019: Towards Continuous Quality Control for Spoken Language Corpora. In: *Proceedings of the 14th International Digital Curation Conference* (IDCC19), https://doi.org/10.2218/ijdc.v15i1.601.

Hedeland, Hanna and Ferger, Anne 2020: Towards Continuous Quality Control for Spoken Language Corpora. *International Journal for Digital Curation*, 15 (1). https://doi.org/10.2218/ijdc.v15i1.601.

ISO/TC 37/SC 4. 2016: *Language resource management* – Transcription of spoken language. Standard ISO 2462:2016, International Organization for Standardization, Geneva, CH. http://www.iso.org/iso/cataloguedetail.htm?csnumber = 37338.

Jettka, Daniel and Lehmberg, Timm 2020: Towards Flexible Cross-Resource Exploitation of Heterogeneous Language Documentation Data. *Proceedings of the 12th Language Resources and Evaluation Conference*, 2901–2905.

Lehmberg, Timm 2020: Digitale Edition des Kuzmina Archivs. *Finnisch-Ugrische Mitteilungen* 44: 121–130.

Sanjek, Rogier (ed.) 1990: *Fieldnotes*. The Makings of Anthropology. Ithaca, London: Cornell University Press.

Sanjek, Rogier and Tratner, Susan (eds.) 2016: *Fieldnotes*. The Makings of Anthropology in the digital world. Philadelphia: University of Pennsylvania Press.

Schmidt, Thomas and Wörner, Kai 2014: EXMARaLDA. In Durand, Jacques, Gut, Ulrike, and Kristoffersen, Gjert (eds.): *Handbook on Corpus Phonology*. Oxford: Oxford University Press, 402–419.

Szeverényi, Sándor and Wagner-Nagy, Beáta 2002: The History of Samoyed Toponymic Research *Onomastica Uralica* 2: 253–259.

Wagner-Nagy, Beáta, Szeverényi, Sándor, and Gusev, Valentin 2018: User's Guide to Nganasan Spoken Language Corpus. *Working Papers in Corpus Linguistics and Digital Technologies: Analyses and methodology* Volume 1. https://ojs.bibl.u-szeged.hu/index.php/wpcl/issue/view/810

Wagner-Nagy, Beáta and Arkhipov, Alexandre 2019: *INEL Bibliographie* (Version 1.0) [Data set]. http://doi.org/10.25592/uhhfdm.731.

DOI: https://doi.org/10.14232/wpcl.2022.7.5

Ursachen und Folgen des Bedarfs nach individuellen Softwarelösungen in den digitalen Geisteswissenschaften am Beispiel der bayerischen Dialektwörterbücher der Bayerischen Akademie der Wissenschaften

Manuel Raaf

1. Einleitung

Die oftmals speziellen und sehr auf das jeweilige Thema ausgerichteten Anforderungen geisteswissenschaftlicher Projekte an die IT verlangen meist auch nach speziellen Softwarelösungen, da bereits bestehende Programme diese Ansprüche nicht bedienen können. Der späte Wechsel bei (insbesondere Langzeit-) Projekten vom analogen zum digitalen Workflow ist ein Grund hierfür, da die gewohnten Arbeitsabläufe meist möglichst beibehalten werden sollen. Hinzu kommt die Anforderung, der breiten Öffentlichkeit online einen niedrigschwelligen Zugang zu Forschungsdaten und -ergebnissen zu ermöglichen. Sowohl die Informationsverarbeitung (Funktionalität) als auch die Informationsdarstellung (Layout) ist daher eine nicht zu unterschätzende Herausforderung, die erfahrungsgemäß mit eigens entwickelter Software besser zu bewerkstelligen ist, als mittels bereits existenter und ggfs. gar generischer. Selbiges gilt für Datenformate/-strukturen.

Anhand der Beispiele aus den Mundartprojekten der BAdW (Bayerisches Wörterbuch, Fränkisches Wörterbuch, Digitales Informationssystem von Bayerisch-Schwaben), der letztjährig übernommenen Bayerischen Dialektdatenbank sowie dem Derivat dieser vier Datenbasen namens *Bayerns Dialekte Online* soll unter Fokussierung der Informationsdarstellung skizziert und diskutiert werden, welche Ursachen und Folgen individuelle Softwarelösungen aufweisen und welchen zusätzlichen Mehrwert sie gegenüber generischen Lösungen liefern können.

2. Ursachen & Lösungen

2.1 Allgemein

Die primären Ursachen, die eine individuelle Softwarelösung erfordern, sind i.d.R. spezifische Eigenschaften des jeweiligen Projekts bezüglich dessen Daten und/oder Arbeitsabläufen. Oft sind die Daten (Projektinhalte) sehr granular erfasst, vielschichtig, komplex und sollen entsprechend den Anforderungen der Projektmitarbeiter:innen individuell verarbeitet sowie dargestellt werden.

Auch der niedrigschwellige Zugang für die interessierte Öffentlichkeit spielt eine zunehmend wichtigere Rolle. Für Neu- sowie Nachfolgeanträge ist zudem die Verwendung informationstechnologischer Methoden eine Voraussetzung für die erfolgreiche Antragsstellung, jedoch erwarten Evaluationsgremien bestehender Langzeitprojekte nicht zuletzt auch deshalb eine entsprechende digitale Umsetzung bisheriger Forschungsdaten und -ergebnisse (vgl. BMBF 2016; DFG 2020; Volkswagenstiftung 2021).

Liegen die Forschungsprimärdaten analog vor in Form von Fragebögen oder Zettelkästen, wie dies bei älteren variationslinguistischen Vorhaben die Regel ist aufgrund der zu Projektbeginn fehlenden

Nicole Palliwoda (ed.): *Data Processing and Visualization in Variational Linguistics/Dialectology*. Working Papers in Corpus Linguistics and Digital Technologies: Analyses and methodology- Vol.7. 2022.

technischen Möglichkeiten, findet man für gewöhnlich keine zufriedenstellende fertige Softwarelösung, die die Struktur der Daten nach deren Digitalisierung entsprechend genau behandeln kann. Hat man bereits digitale Daten in Form von Word-Dateien, sind diese jedoch nur bedingt maschinenlesbar und bedürfen einer Konvertierung, die sich stark an die im Projekt verwendeten Formatierungen halten muss. Fehlt es diesen an Eindeutigkeit, ist eine Umwandlung ohne manuelle Korrekturarbeiten ohnehin nicht möglich. Sollen Projekte vernetzt werden, die unterschiedliche Datenqualitäten bereitstellen (z. B. redaktionell verfasste Artikel vs. Beleglisten) und/oder verschiedene Datenformate nutzen (SQL vs. XML vs. Text mit individuellen Metazeichen vs. Word vs. Print-PDF), kann dies auch nur von eigener Software bzw. zumindest eigenen Modulen für existente Software gelöst werden und bedeutet damit den Bedarf an individueller Programmierung. Fremdsoftware stellt nämlich hierfür i.d.R. nicht (die gesamte) nötige Funktionalität bereit. Bedenkt man die Spezifika des eigenen Projekts, ist dies auch nicht mehr verwunderlich: Fremde Programme können die projekteigenen Strukturen selbstredend nicht kennen oder ausreichend flexibel verarbeiten, selbst wenn es sich um Software aus thematisch ähnlichen Projekten handelt (z. B. innerhalb des Bereichs der Dialektlexikographie).

Zwar gibt es seit Beginn der Digital Humanities Bestrebungen, all diese Problemfälle im jeweiligen Anwendungsbereich oder gar darüber hinaus generisch in einer Softwarelösung abzufangen, doch mangels entsprechender künstlicher Intelligenz kann Software dies aktuell schlichtweg noch nicht leisten. Weitgehend generisch konzipierte Lösungen bleiben letztlich Insellösungen (BBAW 2020; Textgrid; Mehler – Gleim 2016)[1] oder scheitern (LRZ 2019). Computer sind durchaus dem Menschen überlegen, wenn es um lineare Rechenaufgaben geht. So können die Inhalte tausender Bücher in wenigen Sekunden durchsucht oder auch konvertiert werden. Auch Bilder können schnell verarbeitet und z. B. zurechtgeschnitten werden. Abstrakte Vorgänge hingegen sind eine immense Herausforderung für die Informationstechnologie, weniger jedoch für den Menschen: Die semantische Disambiguierung von Homonymen ist für uns ein Leichtes im direkten Vergleich zur algorithmischen Lösung, da wir hierfür auf ein entsprechend großes Weltwissen zurückgreifen und dieses logisch adäquat verarbeiten können (vgl. Liqun Luo 2018; Rodriguez-Ramos 2018). Auch die Entscheidung darüber, welche Inhalte verschiedener Ressourcen miteinander verbunden werden müssen für eine quellenübergreifende Suche ist nach wie vor Arbeit, die vom Menschen übernommen werden muss. Software kann hier unterstützen, muss dann jedoch für den jeweiligen Anwendungsfall geeignet und damit spezifisch anstatt generisch sein.

Es bedarf somit der Erstellung eigener Programme, die innerhalb der Institution an die eigenen Belange und somit sehr projektnah ausgerichtet sind, um die genannten Probleme zu lösen. Anwendungen dieser Art werden auch als „In-House-Software“ bezeichnet, da sie speziell und i.d.R. zunächst nur für den hausinternen Gebrauch geschaffen werden. Die Möglichkeit der Verwendung bereits existenter Software sollte im Vorfeld jedoch unbedingt eruiert werden, allerdings immer im Hinblick auf die zur Verfügung stehende Zeit: Ein Ausprobieren von Fremdsoftware über Monate hinweg kann den Projektverlauf immens verzögern, wenn sich letztlich und damit (zu) spät herausstellt, dass die Software nicht im Rahmen des Vorhabens eingesetzt werden kann.

[1] Die hier beispielhaft genannten, sehr großen und zurecht bekannten Projekte bedienen mehrere Arbeitsbereiche innerhalb der digitalen Geisteswissenschaften, können doch auch innerhalb dieser nicht ohne projektspezifische Erweiterungen wirklich alle Bedürfnisse aller Projekte generisch abdecken. Somit stellen auch sie Insellösungen dar, wenn auch recht große.

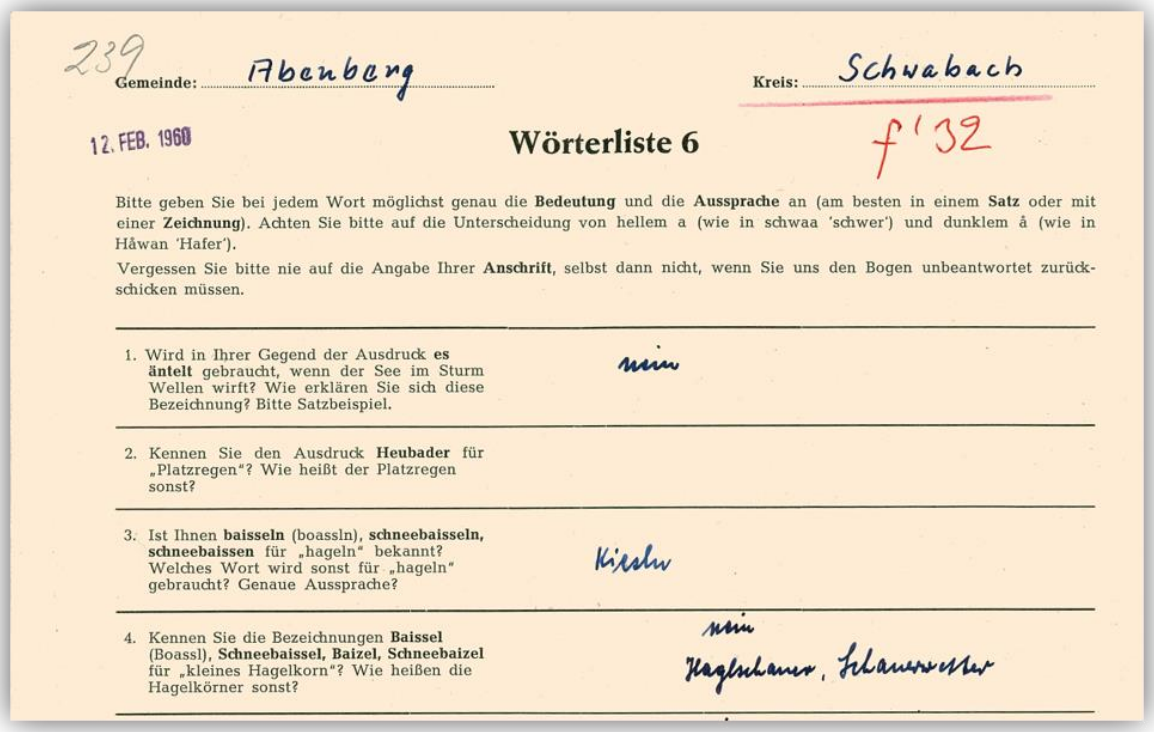
2.2 Das Bayerische Wörterbuch (BWB)

2.2.1 Ursachen

Das Bayerische Wörterbuch führte ab 1913 verschiedene Erhebungen durch, von denen bis auf die Wörterlisten, die 1958 begannen, alle abgeschlossen sind (Schnabel et al. 2020: 50ff). Die allesamt in Papierform und in unterschiedlichen Formaten vorhandenen Rückläufe dieser Erhebungen stellen den primären Quellenbestand des BWB dar. Sie wurden bis 2016 händisch ausgewertet, um als Belegmaterial in die in Microsoft Word geschriebenen Artikel einfließen zu können. Eine unterstützende digitale Komponente existierte bis dato nicht im Arbeitsablauf.

Allein für die sehr umfangreiche Sammlung der Wörterlisten existieren rund 100.000 Exemplare mit jeweils vier Seiten, auf denen 60 Fragen enthalten sind (siehe Abbildung 1). Durch die Vielzahl an Rückläufen aus dem Untersuchungsgebiet ergeben sich über alle Sammelorte hinweg insgesamt etwas mehr als sechs Millionen einzelne Belege innerhalb dieser Sammlung. Die Anforderung an die digitale Unterstützung war daher, Belege gezielt nach Bogen, Frage und/oder Verortung aufrufen und mit wenigen Mausklicks einem Lemma zuweisen zu können.

Abbildung 1 *Ausschnitt eines Fragebogens*



239 Gemeinde: Abenberg Kreis: Schwabach

12. FEB. 1960

Wörterliste 6 f 132

Bitte geben Sie bei jedem Wort möglichst genau die **Bedeutung** und die **Aussprache** an (am besten in einem **Satz** oder mit einer **Zeichnung**). Achten Sie bitte auf die Unterscheidung von hellem a (wie in schwaa 'schwer') und dunklem å (wie in Håwan 'Hafer').

Vergessen Sie bitte nie auf die Angabe Ihrer **Anschrift**, selbst dann nicht, wenn Sie uns den Bogen unbeantwortet zurückschicken müssen.

1. Wird in Ihrer Gegend der Ausdruck **es äntelt** gebraucht, wenn der See im Sturm Wellen wirft? Wie erklären Sie sich diese Bezeichnung? Bitte Satzbeispiel. — nein

2. Kennen Sie den Ausdruck **Heubader** für „Platzregen"? Wie heißt der Platzregen sonst?

3. Ist Ihnen **baisseln** (boassln), **schneebaisseln, schneebaissen** für „hageln" bekannt? Welches Wort wird sonst für „hageln" gebraucht? Genaue Aussprache? — Kiesln

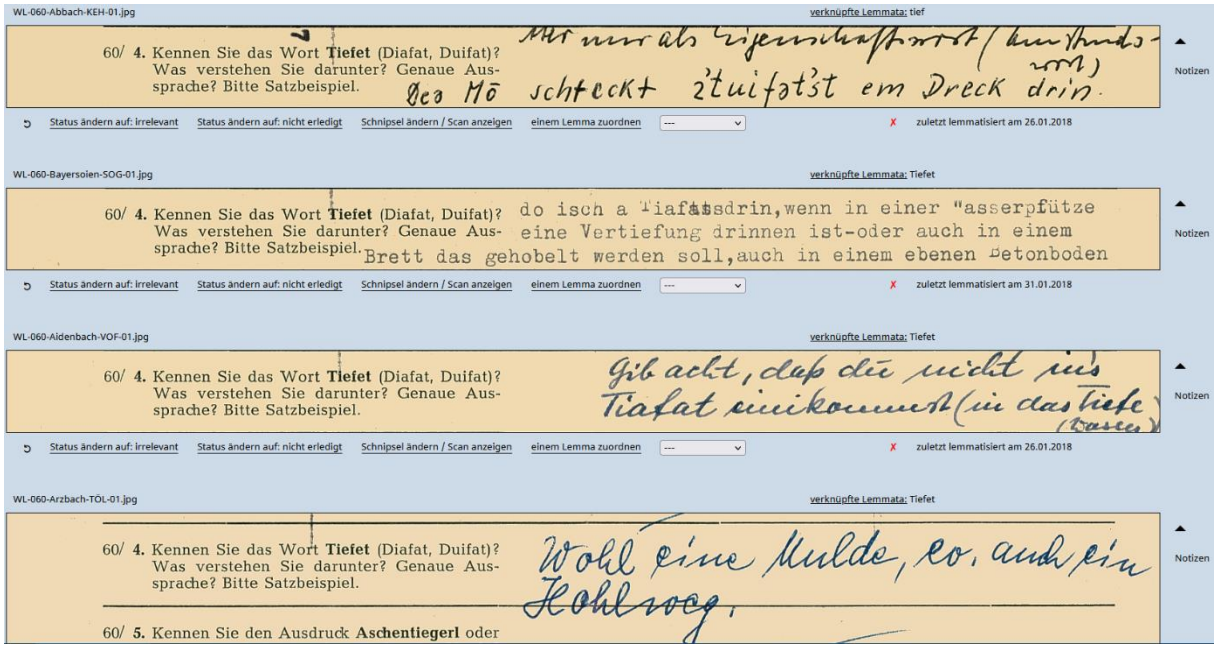
4. Kennen Sie die Bezeichnungen **Baissel** (Boassl), **Schneebaissel, Baizel, Schneebaizel** für „kleines Hagelkorn"? Wie heißen die Hagelkörner sonst? — nein Hagelkorn, Schauerkorn

2.2.2 Lösungen

Im April 2016 begann die digitale Unterstützung der redaktionellen Arbeitsabläufe am BWB, indem das für das Fränkische Wörterbuch entwickelte Redaktionstool in angepasster Form für das BWB in Betrieb genommen wurde (Raaf 2016a; siehe auch 2.4.2). Wörterbuchartikel schrieb die Redaktion zunächst weiterhin mit Microsoft Word; das Tool erleichterte lediglich die Exzerpierung. Hierfür wurden die Wörterlisten in einzelne Bildschnipsel zerlegt (Raaf 2017), sodass über diese gezielt eine bestimmte Frage eines Bogens aufgerufen werden konnte. Irrelevante Belege (d.h. leer oder ohne fragenbezogenen

Inhalt) können zudem vorher sehr effizient durch Hilfskräfte markiert werden, um in der Suche ausgefiltert zu werden (siehe Abbildung 2). Bereits vergebene Lemmata lassen sich über eine Auswahlliste mit wenigen Klicks dem Beleg zuordnen, sodass das Exzerpieren mithilfe der projektspezifischen Softwarefunktionalitäten insgesamt um ein Vielfaches schneller vonstattengeht, als im zuvor analogen Betrieb (Schnabel et al. 2020: 70f). Eine lemmatisierte Belegliste lässt sich bei Bedarf in einer PDF zusammenstellen, mithilfe derer die Redaktor:innen dann den Artikel schreiben.

Abbildung 2 *Redaktionstool des BWB*



Im April 2019 erfolgte schließlich nach einigen Monaten der Vorbereitung der Umstieg auf einen komplett digitalen Arbeitsablauf, indem Artikel seither mittels Oxygen[2] in das zuvor in engem Kontakt mit der Redaktion ausgearbeitete XML geschrieben werden (vgl. Abbildung 3 und Abbildung 4). Die Entscheidung, hierfür ein eigenes XML anstatt des De-Facto-Standards TEI zu verwenden, fiel aufgrund pragmatischer Gründe: Die Interoperabilität, die für TEI einen der oft genannten Hauptgründe darstellt, geht verloren, wenn ein Derivat erstellt wird, um die projektspezifischen Strukturen abbilden zu können. Ohne die Übermittlung des zugehörigen Schemas ist damit ein solches TEI-Derivat nicht automatisch interoperabel mit anderen, ggfs. nur das originale TEI verwendenden Projekten. Das Bereitstellen des Schemas lässt hingegen den diesbezüglichen Unterschied zwischen dem Derivat und einem eigenen XML verschwinden.[3] Darüber hinaus ist TEI in seiner Basisform ebenso wie das auf die Lexikographie bezogene Derivat „TEI Lex-0“ (vgl. Bański et. al 2017; vgl. DARIAH-ERIC Working Group

[2] Oxygen ist ein weit verbreiteter, leistungsfähiger, jedoch kostenpflichtiger XML-Editor.

[3] Wer TEI gewohnt ist, wird sich in einem weitgehend am ursprünglichen Format orientierten Derivat selbstredend schneller zurechtfinden, als in einem völlig davon abweichendem. Ob und wie weit das Derivat diese Anforderung jedoch erfüllt, ist völlig frei derjenigen Person überlassen, die es erstellt. Es kann daher nicht pauschal gesagt werden, dass TEI-Ableitungen generell leichter nutzbar sind als eigene XML-Strukturen. Aus diesem Grund sind nach Ansicht des Autors beide Lösungen in Bezug auf die Interoperabilität als grundsätzlich gleichwertig anzusehen.

2017) aufgrund seines jeweils ganz eigenen Vokabulars von dem des Arbeitsalltags der Redaktion weit entfernt.

Die Verwendung eigener Schemata und gar sogenannter domänenspezifischer Sprachen erwies sich darüber hinaus auch in einem anderen an der BAdW angesiedelten Projekt bereits als sinnvoll (vgl. Arnold 2019; Arnold 2020; Arnold 2021): Im Mittellateinischen Wörterbuch ist es durch die Verwendung eigener Lösungen bezüglich des Formats, das in einem einfachen (und dazu noch kostenfreien) Texteditor eingegeben werden kann, möglich, zum einen sehr auf die Wünsche der Redaktion einzugehen und damit selbst hochkomplexe Anforderungen an die Struktur vergleichsweise einfach redaktionsnah abbilden zu können. Zum anderen ist es möglich, dieses Format durch eigene Konverter in jedes andere zu übertragen und somit letztlich auch den Drucksatz selbst zu produzieren, z. B. mittels LaTeX[4], sowie eine HTML-Ausgabe für die Online-Publikation zu erzeugen. Auf Autovervollständigung, die Einbindung von kontrolliertem Vokabular, oder anderen arbeitserleichternden Eingabehilfen muss während der redaktionellen Arbeit nicht verzichtet werden, sodass dieser Ansatz eine vollständige Alternative zum XML-Editor darstellt. Für das BWB war jedoch eine domänenspezifische Sprache nicht erforderlich, sodass es bei einem redaktionseigenen XML belassen wurde (vgl. Abbildung 4).

[4] LaTeX ist eine freie Software, die das freie und leistungsstarke Textsatzsystem TeX verwendet.

Abbildung 3 *Artikel „predigen“ des BWB im Word-Format*

predigen

Vb. **1** die Predigt halten, (Gottes Wort) verkündigen.— **1a** (im Gottesdienst) predigen, °Gesamtgeb. vielf.: *pretit* „gepredigt“ Reitrain MB; *wans schee prödöngan, nacha is dö ganz Kiacha voi Lait* Hengersbg DEG; *da Här Pfårra tuat bretinga* Rgbg; *Vom Eh'stand hat der Pfarra 'predigt* {Fliegende Bl. (München) 73 (1880) 99}; *konnst du mia vielleicht erzähln, was da Pfarrer predigt hod?* {HERRLEIN Wallfahrt 15}; *er prediget an sand Matheiss tag* {ARNPECK Chron. 539,12}; *nutzt kein bredigen auch nicht* {Bilanz 1782 26}.— Ra.: *der Pfarrer predigt nicht zweimal / nur einmal* u.ä. Weigerung, etwas Gesagtes zu wiederholen, °OB, °NB vielf., °OP mehrf., °MF, °SCH vereinz.: °*moanst du laichd, da Bfara bredigd dswoamoi fia di!* Rosenhm; °*da Pfoarra predicht blouß oamal* Wdsassen TIR; „*Der Pfarrer predigt nicht zweimal …* wenn einer nicht aufpaßt, wenn man ihm etwas sagt“ {Oberpfalz 78 (1990) 194}.— **1b** (Gottes Wort) verkündigen: *dös werd scho' dö richtige Religion sei', dö wo der predigen kunnt* {LUTZ Zwischenfall 38}; *Categorizo ich #p³[re]dion* Tegernsee MB 11.Jh. {StSG. IV,242,11}; *do er sein hilig junger daz hilig ewangelium in der heidenscheft #u⁶ber alle die werlt bredigen hiezze* {O'altaicher Pred. 29,8-10}; *es sei dann in derselben Woche kein Feiertag, daran göttlichs Wort gepredigt* Wunsiedel 1544 {ZILS Handwerk 23}. **2** in eindringlicher Weise ermahnen, ans Herz legen, OB, NB vereinz.: *hör auf mit deim Predign!* Haag WS; *Das is die christli' Menschenlieb, woaßt, die si uns predinga* {S. SCHUBAUR, Mein Vermächtniß an Bayern, Leipzig 1831, 294}; *Da sitzt aa so oana, der allawei predigt hat: nur Ruhe – nur Ruhe!* {THOMA Werke II,256 (Lokalbahn)}.— Auch in fester Fügung: *moralpredign* eine Moralpredigt halten G'weismannsdf FÜ.
3: °*predign* „schnurren, von der Katze“ Parsbg.

Abbildung 4 *Artikel „predigen“ des BWB im XML-Format*

```
<bdo>
  <artikel id="bwb_predigen" wb="bwb">
    <lemma-position>
      <lemma>predigen</lemma>
      <grammatik wortart="Vb" numerus="Singular"/>
    </lemma-position>
    <bedeutung-position nr="1">
      <bedeutung>die Predigt halten, (Gottes Wort) verkündigen</bedeutung>
      <bedeutung-position nr="1a">
        <bedeutung>(im Gottesdienst) predigen</bedeutung>
        <verbreitung-position>
          <verbreitung-angabe belegt="nach1958" bezirk="Gesamtgebiet" häufigkeit="vielfach"/>
        </verbreitung-position>
        <beleg-position>
          <beleg-angabe>
            <beleg-text>pretit</beleg-text>
            <beleg-kontext provenienz="zitat">gepredigt</beleg-kontext>
            <beleg-quelle>
              <beleg-region ort-landkreis="Reitrain-MB"/>
            </beleg-quelle>
          </beleg-angabe>
          <beleg-angabe>
            <beleg-text>wans schee prödöngan, nacha is dö ganz Kiacha voi Lait</beleg-text>
            <beleg-quelle>
              <beleg-region ort-landkreis="Hengersberg-DEG"/>
            </beleg-quelle>
          </beleg-angabe>
          <beleg-angabe>
            <beleg-text>da Här Pfårra tuat bretinga</beleg-text>
            <beleg-quelle>
              <beleg-region ort-landkreis="Regensburg-R"/>
            </beleg-quelle>
          </beleg-angabe>
          <beleg-angabe>
          <beleg-angabe>
          <beleg-angabe>
          <beleg-angabe>
          <phraseologie-position>
            <phraseologie>
              <phraseologie-typ>Ra.</phraseologie-typ>
              <phrase>
                <beleg-typ>der Pfarrer predigt nicht zweimal / nur einmal</beleg-typ>
                <erklaerung>u.ä.</erklaerung>
                <bedeutung>Weigerung, etwas Gesagtes zu wiederholen</bedeutung>
                <verbreitung-position>
                  <verbreitung-angabe häufigkeit="vielfach" belegt="nach1958" bezirk="OB"/>
                  <verbreitung-angabe häufigkeit="vielfach" belegt="nach1958" bezirk="NB"/>
                  <verbreitung-angabe belegt="nach1958" bezirk="OP" häufigkeit="mehrfach"/>
                  <verbreitung-angabe häufigkeit="vereinzelt" belegt="nach1958" bezirk="MF"/>
                  <verbreitung-angabe belegt="nach1958" bezirk="SCH" häufigkeit="vereinzelt"/>
                </verbreitung-position>
                <beleg-angabe belegt="nach1958">
                  <beleg-text>moanst du laichd, da Bfara bredigd dswoamoi fia di!</beleg-text>
                  <beleg-quelle>
                    <beleg-region ort-landkreis="Rosenheim-RO"/>
                  </beleg-quelle>
```

2.3 Das Dialektologische Informationssystem von Bayerisch-Schwaben (DIBS)

2.3.1 Ursachen

Im August 2017 kam das Dialektologische Informationssystem von Bayerisch-Schwaben an die Bayerische Akademie der Wissenschaften. Es brachte eine bereits seit vielen Jahren befüllte Access-

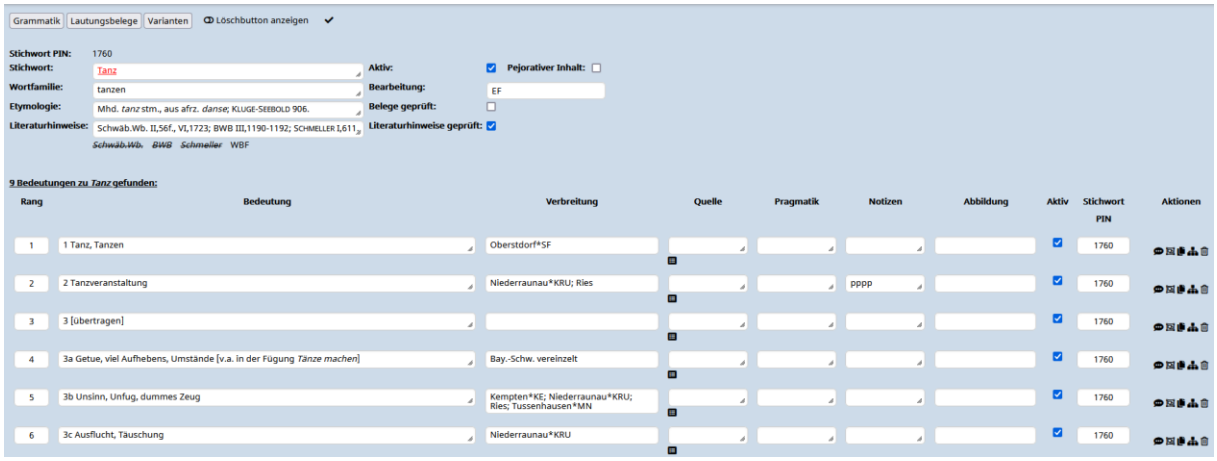
Datenbank mit, die die Grundlage für das 2013 erschienene Dialektwörterbuch von Bayerisch-Schwaben darstellte (Schwarz 2013). Digitalisate von Fragebögen waren in dieser Datenbank nicht eingebunden, sollten aber im Laufe der Zeit hinzukommen. Da ein kollaboratives Arbeiten ein einer einzelnen und ohne Kontrollmechanismen versehenen Access-Datei weder komfortabel noch fehlerfrei möglich war, bedurfte es einer Lösung, die dies ermöglichte und zudem die bisher nur offline vorhandenen Daten auch ins Web stellen konnte. Dabei sollte der bisherige Arbeitsablauf jedoch weitgehend beibehalten werden können – auch bez. der Eingabe und der dafür notwendigen Anzahl an Mouse-Klicks –, sodass die Lösung eine Art Klon der bisherigen Access-Formulare werden musste.

2.3.2 Lösungen

Anfang 2019 wurde deshalb ein sehr am Aufbau der bisher genutzten Access-Datenbank orientiertes, webbasiertes Redaktionstool entwickelt (Funk et. al 2020; Raaf 2020). Zunächst wurde es nur redaktionsintern verwendet, um die Artikelinhalte in gewohnter Manier einzugeben. 2020 erfolgte schließlich die Freischaltung für die interessierte Öffentlichkeit. Im weiteren Projektverlauf soll die Anwendung mit Digitalisaten, Audio- und Videoaufnahmen und Vernetzungen mit anderen Projekten aufgewertet werden. Zu diesem Zwecke werden bis Herbst 2021 zwei Typen von digitalisierten Fragebogenerhebungen in Schnipsel von Einzelbelegen aufgeteilt und lemmatisiert, sodass diese – wie auch im BWB und WBF – gezielt aufgerufen werden können.

Eine bereits bestehende und für den Anwendungsfall allumfassend passende Lösung, die Access-Datenbank mitsamt ihrer Formulare online zu stellen und im Zugriff der öffentlichen Ansicht auf Leserechte zu beschränken, existiert nach Wissen des Autors nicht. Zwar bietet Access die Möglichkeit an, Webdatenbanken zu erstellen, jedoch sind diese im Funktionsumfang eingeschränkt.

Abbildung 5 *Redaktionstool des DIBS*



2.4 Das Fränkische Wörterbuch (WBF)

2.4.1 Ursachen

Das Fränkische Wörterbuch entstand 1933 als „Ableger des Bayerischen Wörterbuchs“ (König/Raaf 2020: 78f). Seit 2004 werden Belege, die in den sogenannten Nachkriegsbogen zwischen 1960 und 2001 erhoben wurden, in Excellisten zusammengetragen und lemmatisiert. Bis zur Entwicklung eines Redaktionssystems, in dem die Fragebögen gezielt durchsucht und angezeigt werden können, geschah das Prüfen der Bögen in manueller Handarbeit. Um z. B. die Frage 4 des Bogens 50 bearbeiten zu

können, mussten hierfür alle Papierbögen aus den Schubern geholt, ggfs. nach Ort und Landkreis sortiert und dann in einzelnen Arbeitsschritten abgetippt, lemmatisiert und annotiert werden. Das Erfassen der Belege konnte nicht durch OCR oder ähnliche Techniken übernommen werden, da die Fehlerrate bei handschriftlich ausgefüllten Listen – insbesondere bei solchen in Sütterlin – zu hoch ist. Die Idee für die digitale Unterstützung war daher Anfang 2015 zunächst einmal nur, die Bildbereiche der Belege jeweils einzeln aufrufen zu können und die dafür erfassten Daten der Excelliste online darstellen zu können

Die verschiedenen Arbeitsschritte in der Digitalisierung der Belege (König – Raaf 2020: 84ff) gestalteten es zudem als äußerst zeitaufwändig, die durch Hilfskräfte und Freiwillige zusammengetragenen Belege aufzuteilen und zu lemmatisieren. Im Frühjahr 2020 kam daher die Anfrage, ob dies vereinfacht werden könnte: Aus z.B. dem Einzelbeleg „nimm a Stickla Brut mit“ und dessen Umschrift sollte je eine Zeile mit der jeweiligen in der Umschrift enthaltenen Grundform (hier also: „nimm“, „ein“, „Stücklein“, „Brot“, „mit“) werden. Dieser Vorgang wurde bis dato händisch durch Kopieren und Einfügen der Zeile sowie dem manuellen Eintippen des Lemmas vorgenommen – bei je nach Bogen zwischen ca. 15.000 und 22.000 Belegen eine sehr zeitaufwändige Angelegenheit. Auch die Auftrennung der Gesamtliste je Bogen zu Einzellisten je Frage und die Sortierung nach Lemma war aufgrund der Größe der Dateien oftmals schwierig, sodass auch dies übernommen werden sollte. Die Anreicherung der dann erstellten Aufteilung mit Bedeutung und Grammatik sollte darüber hinaus aus der bestehenden Datenbank übernommen werden, sofern Übereinstimmungen vorhanden sind.

2.4.2 Lösungen

Im Früher 2015 wurde mit Entwicklung eines Redaktionstools namens *LexHelfer* begonnen, um der lexikographischen Arbeit der Redaktion ein Helfer zu sein. In diesem sind die Belege alle einzeln aufrufbar, sodass digital und nicht mehr in Papierform exzerpiert werden kann (König – Raaf 2020: 88ff). Es stellt zudem nach dem Upload die Excelliste online dar, um sie granular und über die SQL-Datenbank deutlich schneller durchsuchen und sortieren zu lassen, als dies in einer Excel-Datei möglich wäre, da hier insbesondere bei vielen tausend Zeilen Excel oft träge oder gar nicht mehr reagiert. Im Laufe der Zeit ergaben sich weitere Wünsche, die in das Tool implementiert wurden, sodass inzwischen auch ganze Spalteninhalte ersetzt werden können, Statistiken zu Lemmata, Grundformen und Bedeutungen verfügbar sind, oder nach jedem Import eine Liste der neu hinzugekommenen Lemmata an die Redaktion gemailt wird.

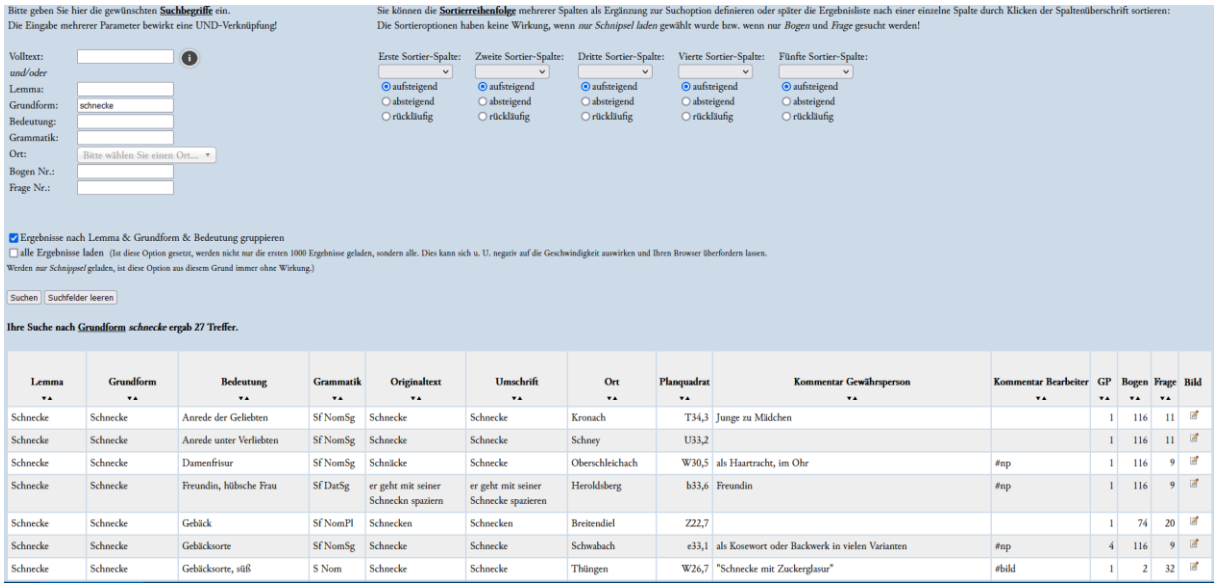
Für die Anfrage, den Lemmatisierungsvorgang in Excel beschleunigen zu können, wurde im Mai 2020 ein VBA-Addon entwickelt, das diese Aufgaben vollautomatisch innerhalb der jeweiligen Exceldatei übernimmt (siehe Abbildung 7).[5]

Die Excellisten online darzustellen und auch zu verarbeiten, ist seit vielen Jahren mit entsprechenden Lösungen von Microsoft oder Mitbewerbern (wie z. B. OnlyOffice) möglich. Sie geben jedoch als webbasierte Implementierungen ihrer Desktop-Varianten nicht die Möglichkeiten, gezielt nur in der Grundform zu suchen und diesen Vorgang an eine weitere Suchebene zu binden, um z. B.

[5] Mit VBA (Visual Basic for Applications), einer Programmiersprache, die von Visual Basic abgeleitet ist, können innerhalb von Office-Dokumenten Programme mitsamt grafischer Benutzeroberfläche erstellt werden, um darüber in z. B. Excel über Formulare den Inhalt des Arbeitsblattes zu bearbeiten oder Dateiinhalte (auch aus dem Web) einzulesen, etc.

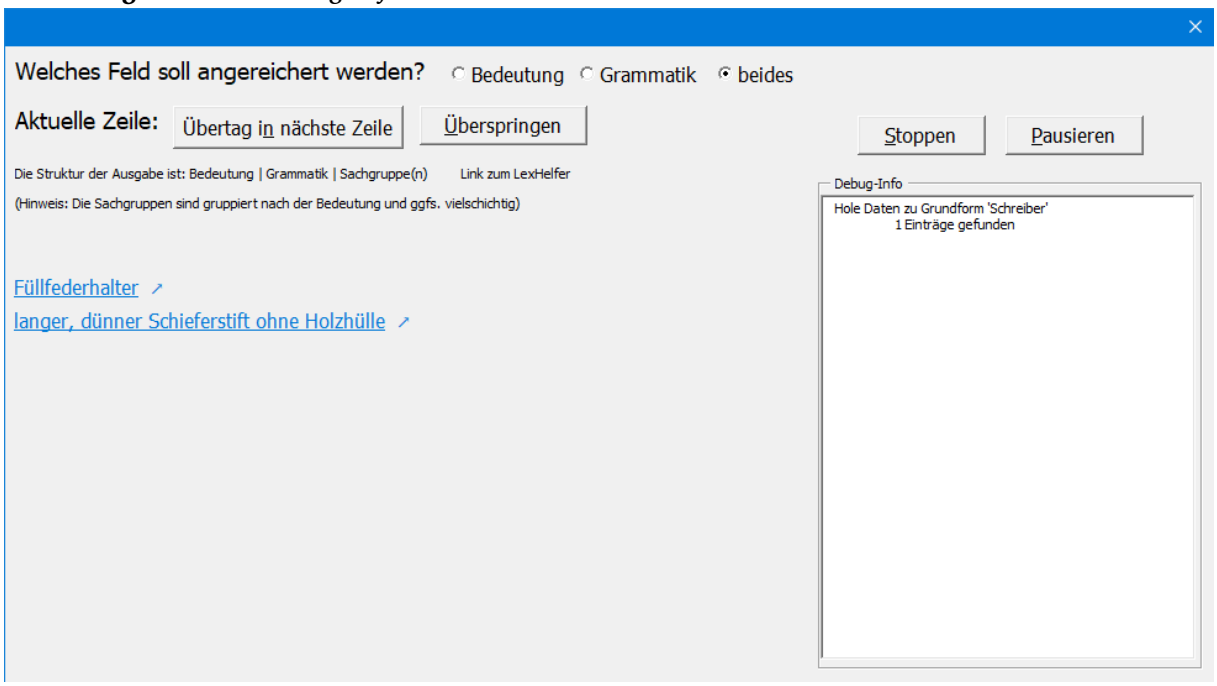
nur Ergebnisse aus bestimmten Landkreisen zu erhalten. Auch die Einbindung von weiterer Funktionalität – wie z. B. der Anzeige der Belegschnipsel oder das erwähnte VBA-Addon – ist in diesen Web-Varianten nicht möglich. Aus diesen Gründen stellte deren Verwendung keine zufriedenstellende Lösung für die Wörterbuchredaktion dar.

Abbildung 6 *Redaktionstool des WBF*



Lemma	Grundform	Bedeutung	Grammatik	Originaltext	Umschrift	Ort	Planquadrat	Kommentar Gewährsperson	Kommentar Bearbeiter	GP	Bogen	Frage	Bild
Schnecke	Schnecke	Anrede der Geliebten	Sf NomSg	Schnecke	Schnecke	Kronach	T34,3	Junge zu Mädchen		1	116	11	
Schnecke	Schnecke	Anrede unter Verliebten	Sf NomSg	Schnecke	Schnecke	Schney	U33,2			1	116	11	
Schnecke	Schnecke	Damenfrisur	Sf NomSg	Schnäcke	Schnecke	Oberschleichach	W30,5	als Haartracht, im Ohr	#np	1	116	9	
Schnecke	Schnecke	Freundin, hübsche Frau	Sf DatSg	er geht mit seiner Schneckn spaziern	er geht mit seiner Schnecke spazieren	Heroldsberg	b33,6	Freundin	#np	1	116	9	
Schnecke	Schnecke	Gebäck	Sf NomPl	Schnecken	Schnecken	Breitendiel	Z22,7			1	74	20	
Schnecke	Schnecke	Gebäcksorte	Sf NomSg	Schnecke	Schnecke	Schwabach	e33,1	als Kosewort oder Backwerk in vielen Varianten	#np	4	116	9	
Schnecke	Schnecke	Gebäcksorte, süß	S Nom	Schnecke	Schnecke	Thüngen	W26,7	"Schnecke mit Zuckerglasur"	#bild	1	2	32	

Abbildung 7 *Lemmatisierungshilfe des WBF*



2.5 Die Bayerische Dialektdatenbank (BayDat)

2.5.1 Ursachen

Die bayerische Dialektdatenbank wurde 2006 veröffentlicht (Zimmermann: 2006) und basierte auf einer durch ein Webinterface abgefragten Oracle-Datenbank, die rund fünf Millionen Belege der bayerischen Sprachatlanten enthielt. Da sowohl die Oracle-Version als auch das Betriebssystem des

zugehörigen Servers nicht mehr aktuell waren, drohte die *BayDat* 2018 im digitalen Nirwana zu verschwinden. Das Rechenzentrum der Universität Würzburg hatte selbstredend keine Kapazitäten für eine Weiter- oder Neuentwicklung des Systems, da i.d.R. kein Rechenzentrum personelle Mittel für umfassende Aufgaben der Art für die gesamte Hochschule bereitstellen kann. Eine Abschaltung der *BayDat* wäre insbesondere für die Dialektwörterbücher der BAdW fatal gewesen, da sie von ihnen sehr regelmäßig konsultiert und als Quelle verwendet wird.

2.5.2 Lösungen

Dank freundlicher Genehmigung der Akademieleitung und des Lehrstuhls für Deutsche Sprachwissenschaft der Universität Würzburg sowie mit Unterstützung der Würzburger Kollegen am dortigen Rechenzentrum wurde die Oracle-Datenbank im Herbst 2018 nach MariaDB konvertiert und eine neue Weboberfläche dafür geschrieben (Raaf 2019).[6] Hierbei wurden die Datenstrukturen angepasst, einige Relationstabellen hinzugefügt, und Fehlerkorrekturen vorgenommen. Mit der Übernahme der *BayDat* eröffnete sich zudem die Möglichkeit, die selten und ausnahmslos in Kapitälchen lemmatisierten Belege vollständig und in üblicher Orthographie zu lemmatisieren sowie mit weiteren Informationen (Bedeutung, Grammatik, Sachgruppe) anzureichern. Das WBF führte dies durch die Förderung des jeweiligen Bezirks für die fränkischen Regierungsbezirke in einem eigens dafür entwickelten Tool aus.

2.6 Projektübergreifende Suche

2.6.1 Ursachen

Zwar sind die Datenbestände von BWB, DIBS, WBF sowie der *BayDat* frei online zugänglich, jedoch nur einzeln durchsuchbar und in der Darstellung der Ergebnisse nicht einheitlich. Im Falle des BWB sind ferner Artikelinhalte nicht granular auffindbar, da sich der Index auf dem Publikationsserver der BAdW (vgl. Bayerisches Wörterbuch 2020) ebenso wie die wortbezogene Suche innerhalb des BWB-*LexHelfers* lediglich auf das Lemma stützt, nicht jedoch auf z. B. Inhalte der Bedeutungspositionen oder der Etymologie. Zudem wurden der jeweilige *LexHelfer* des BWB und des WBF anfangs nur für die interne Nutzung konzipiert, sodass das Erscheinungsbild wenig modern und insbesondere auf mobilen Endgeräten kaum komfortabel erscheint. So entstand 2018 die Idee, eine Aufhebung dieser Heterogenitäten in einem gemeinsamen System zu verwirklichen und dabei Unterschiede der Projektbestände gezielt zu vereinheitlichen. Neben der Homogenisierung waren die Anforderungen insbesondere der niedrigschwellige Zugang, die Zitierfähigkeit von Suchergebnissen über persistente Links, die Visualisierung der Beleginformationen auf einer Landkarte, die Nachnutzbarkeit der Daten und die einfache spätere Erweiterung bzw. Pflege. Eine bestehende Softwarelösung für diese Belange war und ist uns nicht bekannt.

[6] Bei Oracle (genauer: Oracle Database) handelt es sich um ein proprietäres Datenbankmanagementsystem der Firma Oracle Corporation. MariaDB ist ein quelloffenes Datenbankmanagementsystem, das aus seinem Vorgänger MySQL hervorging und inzwischen MySQL – nachdem dieses 2009 von Oracle Corporation übernommen wurde – weitgehend ersetzt hat innerhalb der OpenSource-Community.

2.6.2 Lösungen

Das Sprachinformationssystem *Bayerns Dialekte Online* vereint die Dialektwörterbücher BWB, DIBS, WBF und die *BayDat* in einem Portal. Es wird seit 2018 (Planung) bzw. 2019 (technische Umsetzung) im IT-Referat der Bayerischen Akademie der Wissenschaften entwickelt und befindet sich im Sommer 2021 in der öffentlichen Beta-Phase, nachdem bereits intern durch die Redaktionsmitarbeiter:innen der beteiligten Wörterbücher Tests durchgeführt und das Feedback entsprechend eingearbeitet wurde. Da es sich bei diesem System um ein technologisches wie auch fachwissenschaftliches Desiderat handelt, konnte hierfür keine bereits existente Software verwendet werden.

Abbildung 8 *Startseite der BDO*

Abbildung 9 *Suchfelder der BDO*

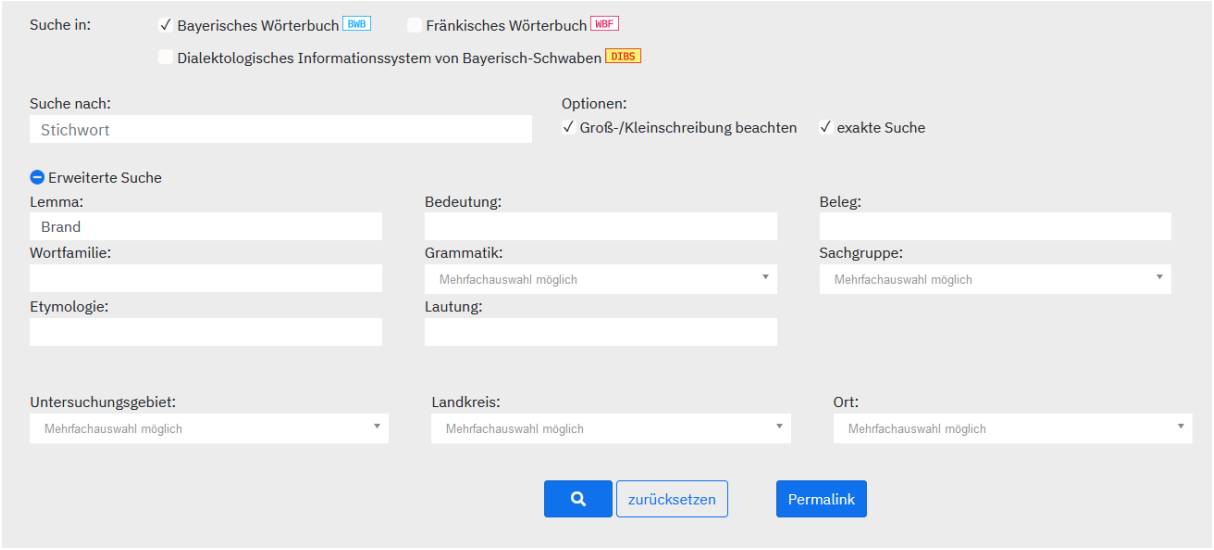


Abbildung 10 *Ausschnitt des Suchergebnisses nach „Brand“*

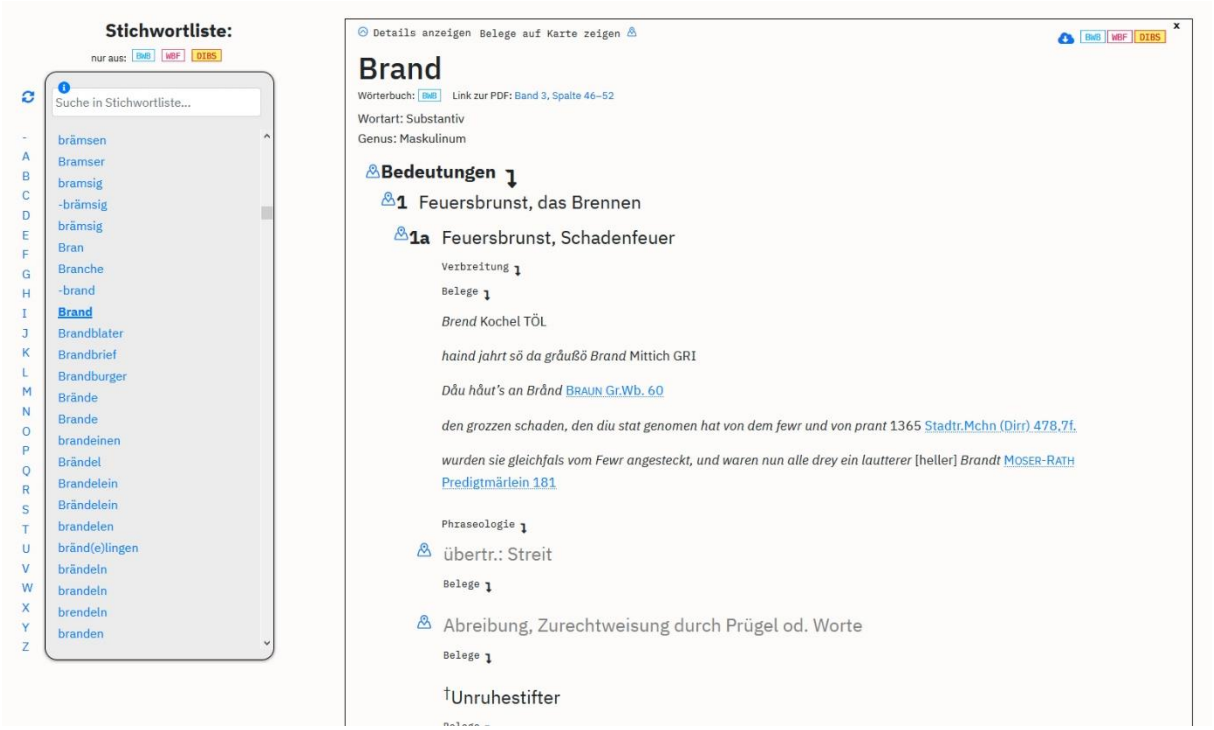
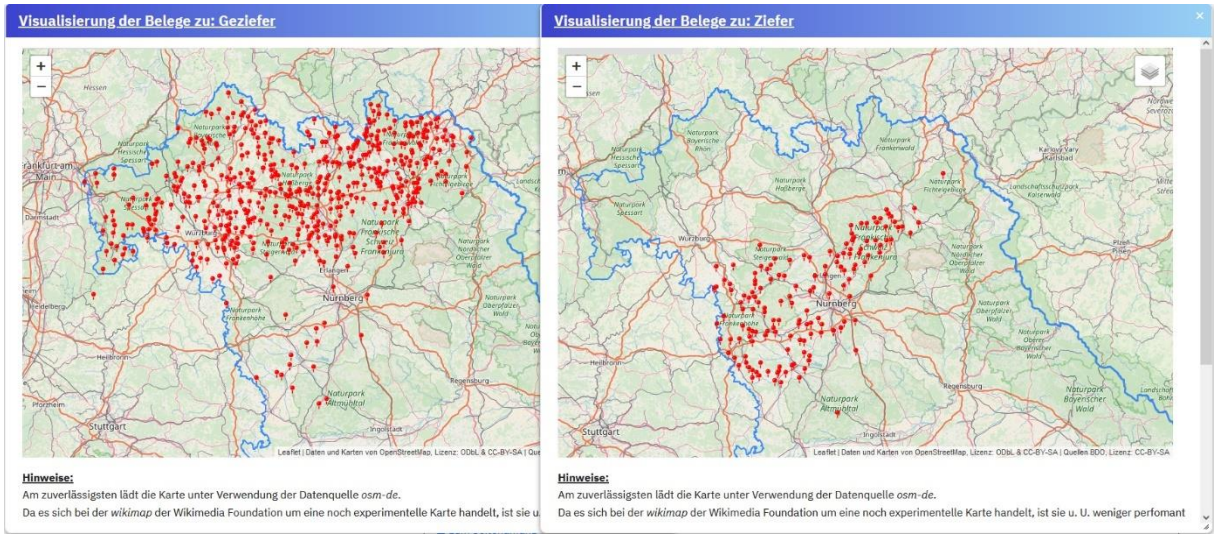


Abbildung 11 *Visualisierung zum Vergleich der Belege von „Geziefer“ mit denen zu „Ziefer“*



3. Folgen

Durch eigene Anwendungsentwicklung ergeben sich neben bedeutsamen Vorteilen ggfs. auch Nachteile, die nachfolgend anhand der Erfahrungen in den illustrierten Projekten wiedergegeben werden. Sie haben jedoch keineswegs Anspruch auf Vollständigkeit, da abhängig vom jeweiligen Projekt weitere auftreten könnten.

3.1 Vorteile

3.1.1 Software nach Maß

Der wohl gewichtigste Mehrwert einer individuellen Softwarelösung ist die Tatsache, dass dadurch ein maßgeschneidertes Produkt entsteht, das für optimale Arbeitsabläufe sorgen kann, und damit den Wissenschaftler:innen eine deutliche Zeitersparnis beschert. Im Falle des BWB z. B. beschleunigte sich das Exzerpieren laut Aussagen der Redaktor:innen um ein Vielfaches durch die Verwendung des Redaktionstools *LexHelfer*.

Auch Darstellung und Vernetzung sind so sehr projektnah und damit individuell realisierbar gemäß den Wünschen der Redaktion.

3.1.1.1 Technologie-Mix

Durch eigene Software können Technologien entsprechend kombiniert werden, um ein Maximum an Nutzen und Flexibilität zu erreichen. Ist man z. B. auf die Verwendung von NodeJS oder PHP allein eingeschränkt, lässt sich sicherlich in den meisten Anwendungsfällen dennoch alles damit realisieren. Jedoch ist das Verwenden anderer Technologien für manche Funktionalitäten ggfs. leistungsfähiger oder schlichtweg einfacher zu implementieren. Im Falle von *Bayerns Dialekte Online* wird ein bunter Mix an Technologien verwendet, um die einzelnen Bereiche bestmöglich umzusetzen, der hier nur exemplarisch genannt werden soll, ohne auf dessen Inhalte nähere einzugehen: Bash, JavaScript, PHP, Python, TeX, XML, XQuery, XSL und SQL (aufgrund des Exports aus den Quelldatenbanken).

3.1.2 Wiederverwendbarkeit

Bedenkt man in der Konzeption der Software bereits, dass sie wiederverwendbar und erweiterbar sein soll, lässt sich hier ein weiterer Vorteil realisieren, der später große Flexibilität ermöglichen kann und die ursprüngliche In-House-Software auch in ähnlichen Projekten außerhalb der eigenen Institution nutzbar werden lässt. Im Falle des *LexHelfers*, der in seiner ersten Variante für das WBF geschrieben wurde, zeigte sich ferner, dass u. U. auch dann eine Wiederverwendbarkeit gegeben sein kann, wenn diese nicht teil der Konzeption ist: Die Schnipselgenerierung sowie -darstellung (Raaf 2017) kann auch für das BWB und alsbald auch für das DIBS verwendet werden.
Eine 100%ige Wiederverwendbarkeit in anderen Anwendungsfällen – noch dazu ohne jeglichen Anpassungsaufwand – bleibt jedoch eine Utopie, da dies bedeutete, eine generische Software mit leistungsfähiger künstlicher Intelligenz geschaffen zu haben.

3.1.2.1 FAIR

Ein bezüglich der Wiederverwendbarkeit besonders wichtiger Aspekt sind die FAIR-Prinzipien (vgl. GoFair 2021): Die Daten sollen auffindbar (*findable*), zugänglich (*accessible*), interoperabel (*interoperable*) und wiederverwendbar (*reusable*) sein, um eine möglichst große Nutzbarkeit und Wiederverwendbarkeit auf allen Ebenen sicherzustellen. In eigener Entwicklung kann die Einhaltung dieser Prinzipien von Anfang bedacht und kontrolliert werden, wohingegen sie in Fremdsoftware erfahrungsgemäß oft an zumindest einem dieser Punkte scheitert. Fairerweise muss jedoch erwähnt werden, dass sich dies in den letzten Jahren gebessert hat, zumindest innerhalb der digitalen Geisteswissenschaften: Moderne Web-Anwendungen, die XML-Exporte zumindest von Suchergebnissen anbieten und mit anderen Anwendungen bzw. Repositorien verknüpft sind, erfüllen zunehmend die FAIR-Prinzipien.

3.1.2.2 Offene Lizenz

Die Frage nach der Lizenz ist zwar nicht zwingend von der verwendeten Software abhängig – sie hängt in erster Linie an etwaigen Urheberrechten der verarbeiteten Daten –, jedoch u. U. von dem von der Software verwendeten Datenformat: Proprietäre Software stellt die verarbeiteten Daten womöglich nicht in einem für die Nachnutzung geeigneten Format bereit. Somit brächte hier eine offene Lizenz nichts, da nichts unter dieser freigegeben werden kann. Doch auch quelloffene Software ist kein Garant dafür, z. B. einen XML-Download anzubieten oder eine Online-Repräsentation als Permalink unter z. B. CC-BY bereitzustellen. Ist die Anforderung danach nicht in der Software implementiert, steht sie selbstredend nicht zur Verfügung. In eigenen Programmen hingegen kann dies sehr einfach gelöst werden und der Datenbestand mit der gewünschten, möglichst offenen Lizenz versehen sowie dadurch letztlich zur Weiterverwendung angeboten werden.

3.2 Nachteile

3.2.1 Hoher Aufwand

Unabhängig davon, ob die Software als Desktopanwendung auf einem lokalen Arbeitsrechner verwendet werden soll oder es sich um eine Webanwendung handelt, fällt insbesondere zu Beginn ein hoher Entwicklungsaufwand an. Die Verwendung von Frameworks und Opensource-Bibliotheken erleichtert und beschleunigt manches sicherlich, jedoch bleibt die Notwendigkeit bestehen, Zeit und

Arbeitskraft zu investieren. Fertige Software benötigt auf der anderen Seite jedoch meist auch einen hohen Einarbeitungsaufwand, da der Zweck einer solchen Software doch sehr speziell ist – schließlich handelt es sich um ein Nischenprodukt innerhalb der (digitalen) Geisteswissenschaften. Müssen zusätzlich eigene Module entwickelt werden, um Funktionalitäten nachzurüsten, vergrößert sich dieser Aufwand entsprechend, da eine fundierte Einarbeitung in den Aufbau der Fremdsoftware vonnöten ist. Eine zusätzliche Gefahr hinsichtlich des Aufwands stellen eigene Erweiterungen zudem aus folgendem Grund dar: Aktualisierungen der Hauptsoftware können die Kompatibilität der Module zu dieser brechen, sodass hier u. U. nach größeren Updates des Hauptprogramms Anpassungen oder vollständig neue Entwicklungen der Module nötig wären.

3.2.2 Softwarepflege

Die Pflege der Software muss ebenfalls bedacht und sichergestellt werden, damit die Entwicklung nicht nach wenigen Jahren Einsatz aufgrund von fehlenden Updates in ihrer Nutzung eingestellt werden muss (siehe hierzu als Beispiel 2.5). Dies stellt in Zusammenhang mit der allseits bekannten, leidlichen Befristungsproblematik einen doppelt zu bewertenden Nachteil dar, da die Pflege von Fremdsoftware nach Weggang des Entwicklers bzw. der Entwicklerin für gewöhnlich nochmals aufwändiger ist. Unter Einhaltung von Entwicklungsstandards ist jedoch zumindest sichergestellt, dass Freelancer über einen Werkvertrag für kleinere Anpassungen im Rahmen von Updates o.ä. gefunden werden können.

3.2.3 Problematische Daten oder Abläufe

Gibt es potentiell problematische Daten im Projekt, stellen auch diese einen nicht zu unterschätzenden Nachteil dar. Es kann sich dabei um Daten handeln, die erst aufwändig aufbereitet oder konvertiert werden müssen, um in die Software eingespeist werden zu können. Es können jedoch auch solche sein, die vielfach fehlerhaft sind: Ortsdaten z. B. sind erfahrungsgemäß nahezu immer problematisch, da sich die Schreibung des Ortsnamens unterscheidet von der offiziellen Schreibweise, Tippfehler enthalten sind, der Altlandkreis verwendet wurde, oder der Ort gar nicht mehr existiert. Die Lösung dieser beiden Problemarten ist jedoch nicht nur für individuelle Software nötig, sondern fiele auch bei vorhandenen Programmen an. Sie stellt somit keinen exklusiven Nachteil für Individualsoftware dar, sollte jedoch unbedingt auch bei dieser bedacht werden.

Neben den Daten selbst ist es auch möglich, dass die Arbeitsweise mit dem geschaffenen Tool nicht optimal abläuft und letztlich eine andere produktiver gewesen wäre. Je nach Projektstand ist ein Wechsel ggfs. (zumindest vorerst) nicht mehr möglich. Doch auch dies ist nicht exklusiv für Individualentwicklungen gültig, sondern kann sich auch nach einigen Monaten der Verwendung von Fremdsoftware einstellen.

3.2.4 Abstimmungsschwierigkeiten

Mit selbstgeschriebener Software eröffnet man Anwender: innen die Möglichkeit, sehr viele Wünsche zu äußern zum Layout und Funktionsumfang. Hier mögen sich eventuell auch nicht alle immer einig sein, sodass es zu Abstimmungsschwierigkeiten kommen kann. Bestehende Software eröffnet hier deutlich weniger oder sogar keinen Diskussionsbedarf, da es schlichtweg keine Spielräume für Wünsche gibt. Im Falle der bayerischen Dialektwörterbücher konnten die Abstimmungs-schwierigkeiten durch kurze Wege und die durchweg gute Kommunikation bisher sehr gut gelöst werden. Lediglich im

LexHelfer des DIBS gibt es vereinzelt Probleme beim Export nach XML, sowie in *Bayerns Dialekte Online* vereinzelt in der Transformation der Datensätze nach HTML. Zwar konnten sie durch Workarounds abgefangen werden, jedoch verkompliziert das den Export bzw. die Transformation an manchen Stellen doch sehr und müsste daher in naher Zukunft durch eine bessere Klärung des Arbeitsablaufes gelöst werden.

4. Fazit

Die genannten Beispiele aus der germanistischen Variationslinguistik zeigen auf, wann und warum individuelle Softwareentwicklungen gegenüber bereits bestehender Software zu bevorzugen sind und worin hierbei die Vor- und Nachteile liegen können. Es wird ebenfalls dargestellt, dass fertige Programme die Nachteile in aller Regel nicht abfangen könnten, sodass sie nicht im logischen Umkehrschluss gegen die Entwicklung von In-House-Software sprechen.

Vor- und Nachteile müssen hierbei immer projektspezifisch abgewogen werden.

Trotz des Plädoyers für Individualentwicklungen hat dieser Artikel keineswegs das Ziel, bereits bestehende Softwarelösungen innerhalb der digitalen Geisteswissenschaften grundsätzlich als ungeeignet darzustellen. Deckt Fremdsoftware die eigenen Bedürfnisse vollends ab, spricht nichts dafür, das Rad neu zu erfinden. Erfahrungsgemäß ist diese Abdeckung allerdings selten gegeben, sodass der sicherere Weg der der Eigenentwicklung ist.

Bibliographie

Arnold, Eckhart 2019– : *Beschreibung der MLW-Notation*. URL: http://purl.badw.de/zy9Hmx#4-die-notation, Stand 22.06.2021

Arnold, Eckhart 2020: *Introduction to DHParser*. URL: https://gitlab.lrz.de/badw-it/DHParser/-/blob/master/Introduction.md, Stand 22.06.2021

Arnold, Eckhart 2021: Ein neues Back-Office für das Mittellateinische Wörterbuch (MLW). In: Bayerische Akademie der Wissenschaften (Hg.): *Jahrbuch 2020*. München: Bayerische Akademie der Wissenschaften. 63-65. URL: http://purl.badw.de/2xfvSh#63, Stand 22.06.2021

Bański, Piotr, Bowers, Jack and Erjavec Tomaž 2017: TEI-Lex0 Guidelines fort he Encoding of Dictionary Information on Written and Spoken Forms. In: Kosem, Iztok, Tiberius, Carole, Miloš Jakubíček, Kallas, Jelena, Krek, Simon and Baisa Vít (Hg.): *Electronic lexicography in the 21st century. Proceedings of the eLex 2017 conference*. Brno: Lexical Computing CZ s.r.o.

Bayerisches Wörterbuch 2020: *Index des Bayerischen Wörterbuchs*. München: Bayerische Akademie der Wissenschaften. URL: https://publikationen.badw.de/de/bwb, Programmierung und Gestaltung von Stefan Müller und Daniel Schwarz, Stand 10.06.2021

BBAW (Hg.) 2020: ediarum. URL: https://www.ediarum.org, Stand 29.06.2021

BMBF 2016: Freier Zugang schafft mehr Wissen. Pressemitteilung. URL: https://www.bmbf.de/de/freier-zugang-schafft-mehr-wissen-3340.html, Stand 10.06.2021

DARIAH-ERIC Working Group 2017: TEI Lex-0.
URL: https://dariah-eric.github.io/lexicalresources/pages/TEILex0/TEILex0.html, Stand 10.06.2021

DFG 2020: Digitaler Wandel in den Wissenschaften. Impulspapier.
URL: https://doi.org/10.5281/zenodo.4191345, Stand 10.06.2021

Funk, Edith, Raaf, Manuel, Schwarz, Brigitte and Welsch, Ursula 2020: „Dialektologisches Informationssystem von Bayerisch-Schwaben (DIBS)“. In: Lenz, Andrea and Stöckle, Philipp (Hrsg.): *Germanistische Dialektlexikographie im 21. Jahrhundert*. Stuttgart: Steiner (Zeitschrift für Dialektologie und Linguistik, Beiheft), 105-142

GoFair 2021: *Fair Principles*. URL: http://www.go-fair.org/fair-principles/, Stand 29.06.2021

König, Almut and Raaf, Manuel 2020: Das Fränkische Wörterbuch (WBF). In: Lenz, Andrea and Stöckle, Philipp (Hrsg.): *Germanistische Dialektlexikographie im 21. Jahrhundert*. Stuttgart: Steiner (Zeitschrift für Dialektologie und Linguistik, Beiheft), 77–104.

Liqun Luo 2018: *Why Is the Human Brain So Efficient*?
URL: https://web.archive.org/web/20210629080632/https://nautil.us/issue/59/connections/why-is-the-human-brain-so-efficient, Stand 10.06.2021

LRZ (Leibniz-Rechenzentrum) 2019: *GeRDI*. Projektbeschreibung.
URL: https://web.archive.org/web/20210601123054/https://www.lrz.de/forschung/projekte/forschung-daten/GeRDI/, Stand 10.06.2021

Mehler, Alexander and Gleim, Rüdiger 2016: *eHumanities* Desktop. Projektwebseite.
URL: https://www.texttechnologylab.org/applications/ehumanities-desktop, Stand 10.06.2021

Raaf, Manuel 2016a: *LexHelfer BWB*. Online-Datenbank der Forschungsprimärdaten des Bayerischen Wörterbuchs. URL: https://lexhelfer.bwb.badw.de, Stand: 10.06.2021

Raaf, Manuel 2016b: *LexHelfer WBF*. Online-Datenbank der Forschungsprimärdaten des Fränkischen Wörterbuchs. URL: https://lexhelfer.wbf.badw.de, Stand: 10.06.2021

Raaf, Manuel 2017: Precise Annotation of Questionnaires for Dialect Research: The Bavarian Dictionary and its Digitization. In: Kosem, Iztok, Tiberius, Carole, Miloš Jakubíček, Kallas, Jelena, Krek, Simon and Baisa Vít (Hg.): *Electronic lexicography in the 21st century. Proceedings of the eLex 2017 conference*. Brno: Lexical Computing CZ s.r.o.

Raaf, Manuel 2019: *BayDat v2*. Die bayerische Dialektdatenbank. URL: https://baydat.badw.de, Stand: 10.06.2021

Raaf, Manuel 2020: *LexHelfer DIBS*. Online-Datenbank der Forschungsprimärdaten des Digitalen Informationssystems zu Bayerisch-Schwaben. URL: https://lexhelfer.dibs.badw.de, Stand: 10.06.2021

Raaf, Manuel 2021: *BDO – Bayerns Dialekte Online*. URL: https://bdo.badw.de, Stand: 10.06.2021

Rodriguez-Ramos, Jaimie 2018: *Brains vs. Computers*.
URL: https://web.archive.org/web/20210629080518if_/https://becominghuman.ai/brains-vs-computers-f769548010f1, Stand 29.06.2021

Schnabel, Michael, Raaf, Manuel and Schwarz, Daniel 2020: Bayerisches Wörterbuch (BWB). In: Lenz, Andrea and Stöckle, Philipp (Hrsg.): *Germanistische Dialektlexikographie im 21. Jahrhundert*. Stuttgart: Steiner (Zeitschrift für Dialektologie und Linguistik, Beiheft), 47–76.

Schwarz, Brigitte 2013: *Dialektwörterbuch von Bayerisch-Schwaben*. Vom Allgäu bis zum Ries. Augsburg: Wißner

TextGrid Konsortium. 2006–2014: TextGrid: Virtuelle Forschungsumgebung für die Geisteswissenschaften. Göttingen: TextGrid Konsortium. URL: https://textgrid.de, Stand 10.06.2021

Volkswagenstiftung 2021: *aufbruch*. Die neue Förderstrategie der Volkswagenstiftung. URL: https://www.volkswagenstiftung.de/sites/default/files/downloads/Aufbruch%20Broschuere_online_2.pdf, 10.06.2021

Zimmermann, Ralf 2006: BAYDAT- Die bayerische Dialektdatenbank. In Schwitalla, Johannes, Stahl, Peter, Wegstein, Werner, and Wolf, Norbert R.: *Würzburger elektronische sprachwissenschaftliche Arbeiten*; Bd. 1. Würzburg: Universitätsbibliothek Würzburg