Digital infrastructure and its impacts on language work: A case study of FieldWorks Language Explorer (FLEx)

Authors

Abstract

Linguists need to collect, organize, analyze, and share data. Despite the variability of linguistic research in terms of inputs and outcomes, there are some common stages, including morphological parsing, lexicon creation, and textual editing. A common software tool for this work is FieldWorks Language Explorer (FLEx). In this article, we examine some of the ways that FLEx's data structures conflict with contemporary practices in language work. These assumptions exclude key groups of users and erect barriers to theoretically and practically relevant research. Where workarounds are feasible, they are both fragile and costly. We use this example to start a broader conversation about software infrastructure for digital linguistic data analysis, and how problems with this infrastructure reflect larger issues in the discipline.

1 Introduction

1.1 Preliminaries

Linguists need ways to collect, organize, analyze, and share data. These processes may be called a project's "workflow", "pipeline", or "data life cycle" (cf. Rehm 2016; Gonzalez et al. 2017; Mattern 2022). Documentation and field methods handbooks dedicate much space to discussion of workflow (Bowern 2015; Chelliah and Reuse 2011; Meakins, Green, and Turpin 2018), and documentation-oriented journals and conferences often include contributions describing new or adapted workflows for fieldwork data collection.

This paper is about the current state of digital infrastructure for data analysis in language documentation and fieldwork. Using Fieldworks Language Explorer (FLEx) software (Summer Institute of Linguistics 2022) as an example, we explore ways in which software design and data structure decisions impact linguistic data workflows, erect barriers to research, and reflect larger problems in language documentation as a discipline.

1.2 What is digital infrastructure?

Many varied language documentation and fieldwork projects have converged on using similar software tools. This digital infrastructure includes applications such as Praat (Boersma and Weenink 2024), ELAN (Wittenburg et al. 2006), and FieldWorks Language Explorer or FLEx (Summer Institute of Linguistics 2022). By software standards, these tools are old -- they have all existed (often in a form very similar to their current one) for at least twenty years. Though minimizing changes to digital infrastructure works in favor of portability, computational uses of language data have expanded enormously over the same period. Increases in processor power, decreases in the cost of storage, and the advent of born-digital video recording have all transformed the possibilities for data collection and annotation. Likewise, the global availability of smartphones has created an almost entirely new set of pipelines for remote or "distributed" fieldwork (Williams et al. 2021; Grzech and Tisalema Shaca 2022; Sneller et al. 2023). Digital data allows for much more extensive computational analysis, which in turn allows us to ask new questions and brings new insights. This includes alignment of text and speech or sign (facilitating nuanced corpus searching); concordancing; part of speech tagging; and dependency parsing, to name just a few.

Alongside these changes in methods, recent work in language documentation also shows a greater awareness of how documentation experts need to respond to the extractive past and present of the discipline. While community-oriented and community-led linguistics has been discussed at least since Wilkins (1992), academic work increasingly also centers Indigenous approaches to language and healing, language advocacy, and language renewal (McCarty et al. 2018; Whalen et al. 2022; Leonard 2017; Woods 2023). These approaches emphasize embedding language in cultural, social, environmental, and artistic practices. They also foreground collaborative practices – where language connects individuals and families with elders and the voices of ancestors – and intergenerational transmission. Scholars in this tradition place a renewed emphasis on reclamation of archival materials, critiquing the aims of salvage documentation and extractive linguistics (Davis 2017; cf. also Tsikewa 2021 for discussion in the context of field methods teaching).

But so far, these changes in the methods and goals of the field have not been matched by changes in its digital infrastructure. We illustrate this through the example of FLEx. This is a software package designed for linguistic analysis and database management. It supports one particular aspect of the fieldwork dataflow – involving the transcription and annotation of text; morphological parsing or glossing; and lexicon building. FLEx is widely used for these processes because it brings together a powerful morphological parser and features that allow the user to compile dictionaries. However, FLEx also makes many – more or less implicit – assumptions about the user and their computer; the language data; the language structure; and the types of output required after analysis. Such assumptions end up facilitating some types of research and impeding others in a way that is problematic for the future of the field. Specifically, as we discuss in more detail below, the assumptions embedded in FLEx exclude key groups of users and erect barriers to relevant theoretical and empirical research.

While we illustrate our arguments using FLEx, our points apply more generally. FLEx is widely used in documentary linguistics and allows us to illustrate multiple points within the same tool. However, our points are not truly about FLEx itself, but about whether the digital infrastructure of linguistics is serving contemporary research needs. All tools embed assumptions in their data structures that inhibit some types of research and facilitate others. That is unavoidable. But in the case of language documentation software, these implicit assumptions deter some users entirely, and force others to develop labor-intensive workarounds which would be unnecessary with a more flexible and linguistically informed software infrastructure. In either case, the end result is that people do less research and applied work which challenges the assumptions embedded in traditional descriptive workflows – for example, they work more with monolingual and monomodal data (§2.2), and collaborate less (§2.3). We highlight these issues for FLEx specifically because of its centrality in morphological analysis workflows and its use in multiple types of research, but similar points can be made about other software.

We begin in §§1.2-1.3 with a discussion of the FLEx software and its aims, before moving in §2 to general issues that shape projects across many linguistic subfields. §3 then describes how FLEx creates barriers to specific use cases.

1.3 What is FLEx?

Fieldworks Language Explorer (FLEx) is a linguistic software suite for data organization and analysis, widely used for analysis of Indigenous language materials. It is produced by the Summer Institute of Linguistics (SIL), a Christian faith-based organization focused on language and Bible translation.[1,2] The software is based on Toolbox, an earlier SIL-produced package with similar functions, and replaced Toolbox in 2006-2007 (Butler and van Volkinburg 2007). FLEx is frequently updated, has always been free to use, and as of v. 9.0 (released September, 2018) is open source.[3]

FLEx combines three main modules: a lexical database, a database of texts, and a morphological parser. Users can gloss texts either by constructing a lexicon and then running the parser, or by

---

[1] From SIL International's mission statement (www.sil.org/about): "Inspired by God's love, we advocate, build capacity, and work with local communities to apply language expertise that advances meaningful development, education, and engagement with Scripture."

[2] SIL has been heavily involved in other multilingual digital infrastructure for many years, through their keyboard tools (such as Keyman), Unicode fonts (such as Gentium), and text rendering tools (XeTeX).

[3] https://software.sil.org/fieldworks/release-notes/

building the lexicon incrementally as they analyze texts (i.e., by training the parser). The lexicon provides structured fields for creating a bilingual dictionary, including a one-way look-up function for finding examples in texts to add to the lexicon. The parsing module allows the use of two algorithms (cf. §3.1): a default string-based parser (XAmple) and a parser that incorporates phonological rules (HermitCrab) .

There are no viable competitors to FLEx that combine all of these functions. Though other packages which link a lexicon to a morphological parser exist, they are generally produced by individual scholars or small teams (Dunham, Cook, and Horner 2014; Sande and Ewert 2023), meaning that they (a) do not offer technical support and (b) may cease to be maintained when their creators shift their professional focus. The closest competitor is ELAN (Wittenburg et al. 2006), a transcription program that provides support for corpus search and has basic interlinearization capabilities. In theory, because ELAN is institutionally supported by the Max Planck Society, it may be a viable competitor. But as of this writing, ELAN's Interlinearization Mode is in beta and lacks key features, such as allowing the user to add new lexical items from the transcript text. ELAN also cannot accommodate text-only data without linked media. These issues mean that ELAN cannot yet compete with FLEx as an interlinearization tool.[4]

Additionally, while collaboration between academic linguists and computer scientists is becoming more common, this trend is unlikely to produce viable alternatives to FLEx. Computer science researchers are simply not interested in developing software for curating small data sets like those used in language documentation. Some teams, including ours, have also experimented with hiring people who have undergraduate-level backgrounds in computer science, and deeper backgrounds in linguistics, to develop project-specific documentation software. While this has sometimes been successful, it is not a long-term solution for the field. The software is too project-specific, and because of funding streams, the developers have too little professional stability to maintain the products. A more feasible long-term solution is to integrate language documentation and revitalization content into degree programs in computational linguistics. This has the potential to create cohorts of computational linguists who both see language documentation data as relevant, and have the technical skills to create alternatives to FLEx.

[4] Several software packages, including Miromaa (Miromaa Aboriginal Language & Technology Centre 2023), Acorns (Harvey 2022), and ILDA (https://mc.miamioh.edu/nbol/ilda-prospective) have been developed for language reclamation work. These place much greater emphasis on the ethical and intellectual property issues discussed in §2. However, none of them have the parsing capabilities of FLEx. Another competitor in the dictionary creation market is TshwaneLex (https://tshwanedje.com/tshwanelex/), a widely used commercial bilingual dictionary program. TshwaneLex has corpus linkage functions but lacks a parser, making it unsuitable for sole use in language documentation.

1.4 FLEx's user aims

Toolbox and FLEx were originally developed for Bible translation,[5] to aid the translator in collecting texts and developing a lexicon from them for use in translation (Dobrin and Good 2009). Today, SIL describes FLEx as "software tools that help you manage linguistic and cultural data." While this description distances FLEx from Bible translation, the training materials provided on SIL's training site[6] still closely integrate FLEx and dictionary training with SIL's Biblical software, such as ParaText.

SIL also states that FLEx is designed for "dictionary development, interlinearization of texts, morphological analysis, and [preparation of] other publications."[7] In line with this description, in language documentation the major functions of the package are to collect language samples – usually called "texts" – and parse them to produce interlinear glossing. Based on the parsed materials, users can compile information about lexicon and morphology and produce bilingual dictionaries in the language of study and the contact language. Thus, the design structure of FLEx is deeply and simultaneously intertwined both with translation as a means and goal of data collection,[8] and with the "Boasian trilogy" of grammar, dictionary, and monologic texts as the primary output of language documentation (cf. Musgrave and Thieberger 2021).

Additionally, FLEx and Toolbox were developed during the discipline's transition from analog to digital fieldwork, and this is evident in their design structure. Analog multimedia data is inherently unlinked except by manual cross-references.[9] In contrast, most digital fieldwork pipelines rely on multidirectional dataflows, where transcriptions can be continuously updated based on analyses and vice versa. This "back and forth" is a major advantage of digital language documentation but is mostly absent in FLEx.

2 General issues: the assumptions of FLEx

FLEx provides solutions for keeping together the entire Boasian trilogy–corpus, lexicon, and grammar. It is the only mainstream language documentation tool that integrates a parser. As a result, items in each module can be linked to each other, and FLEx delivers handily on its promise to help with glossing texts by using the lexicon. Users can integrate data from other sources and also prepare it for use elsewhere. In the 'Lexicon' module, users can add audio and image data and specify variants, multiple senses, and allomorphs. In the corpus (called the 'Texts and Words' module in FLEx), users can perform regular expression concordance searches,

[5] https://web.archive.org/web/20110912212755/www.sil.org/computing/shoebox/TransQA.html

[6] https://lingtran.net/LingTranSoft-Wiki

[7] https://software.sil.org/fieldworks/

[8] It is relevant, but beyond the scope of this article, to note the prominent role in translation theory played by SIL members, most notably Eugene Nida.

[9] For example, once a cassette tape was transcribed, data analysis typically happened only with the transcribed materials, with minimal reference back to the audio information.

automatically gloss text, and add lexemes from within the text view. In the 'Grammar' module, users can add phonological rules that help the parser make hypotheses about what might be an allomorph, as well as take notes on grammatical features as the user learns more about the language of study.

At the same time, because of its origins, FLEx is optimized for certain types of language work. It also makes a variety of assumptions about the user, their computer hardware, and the mono- vs. multilingual content of text. Because these issues potentially affect all of the types of projects described in §3, we review them together. We begin with a review of the technical requirements for running FLEx, then move to the package's treatment of multilingual content, before closing this section by discussing collaboration and integration between language documentation and reclamation.

2.1 Technical requirements and accessibility

The technical requirements of FLEx create a variety of obstacles for users, especially users with lower budgets or less access to computing support. Because FLEx is available only for Windows and Linux, Mac-based researchers cannot run FLEx on their own computers without a virtual machine or remote desktop, and installing a virtual machine requires non-trivial effort and cost. Furthermore, because FLEx requires a disproportionately large amount of RAM, it performs poorly unless run natively on a machine with high RAM. Syncing FLEx databases between computers is also complex and can lead to data loss, as we explore further in §2.3.

Because of these issues, FLEx functions best for a user working on a single Windows computer that has plenty of RAM and a good CPU. As posts to the FLEx Users list hosted by Google Groups frequently show, many actual and potential users of FLEx do not meet this description -- for example, because their hardware does not have enough RAM,[10] they use Macs and have trouble running virtual machines,[11] they need to use multiple computers,[12] or they are collaborating.[13] Some users work around the OS and RAM requirements by purchasing dedicated computers used only for FLEx database work. While this can reduce FLEx-related performance issues on the user's main computer, it is too expensive for many teams. For example, a laptop appropriate for running FLEx might cost $1,000 of a $2,000 seed grant.

These technical issues alone exclude many people from using FLEx for language work. Similar problems also arise from FLEx's lack of visual accessibility. The app is not compatible with screen readers and does not allow users to adjust the text size of the menu items, the text color,

---

[10] https://groups.google.com/g/flex-list/c/IS80wT9twmw/m/9Au36z9cCgAJ

[11] https://groups.google.com/g/flex-list/c/nUMpdlicyn4/m/CQQm8jN3AwAJ, https://groups.google.com/g/flex-list/c/IUYOdhiYyuU/m/casTnueyBgAJ

[12] https://groups.google.com/g/flex-list/c/oCGBZJYIlbM

[13] https://groups.google.com/g/flex-list/c/kxb6Yy93DFM/m/Hs1UzrpzBQAJ, https://groups.google.com/g/flex-list/c/wUmr9gFDvvc, https://groups.google.com/g/flex-list/c/jycmtNbDIVA/m/BarJG0AODAAJ

or background colors.[14] This is especially problematic because the default text size is relatively small and the background colors are bright and low-contrast. This design excludes users who rely on screen readers or color adjustments, and prevents elderly language users with visual impairments, such as cataracts, from viewing FLEx databases. FLEx also has very limited keyboard navigation (most features are accessible only by mouse-click), further limiting its use with screen readers or voice activation. More generally, this lack of accessibility sends negative messages about who is welcome to use the software.

2.2 Mono-content

FLEx is optimized for language data that is *mono-*: monolingual, monolectal, monologic, and monomodal. It works best when the data is homogenous - for example, always grammatical - and the parser functions best on exclusively concatenative morphology. This optimization reflects a set of implicit assumptions about what languages and types of data users want to analyze. When we analyze only homogeneous and "mono" data, these assumptions are not obvious. But when we analyze data that breaches the assumptions, they become clear immediately.

2.2.1 Mono- and multimodal data

FLEx does not support video media, meaning that it cannot be used to analyze signed languages.[15] It also requires all annotations to correspond to a single line of the baseline text. This prevents users from adding information – including as notes – about prosody, gesture, co-occurring activity, or other behaviors that have different time boundaries from the baseline text lines.

2.2.2 Mono- and multilingual research

FLEx texts and databases can be multilingual, but the program's data structure forces each line of a text[16] to be treated as monolingual. This means that, in lines containing intrasentential code-switches out of the study language, researchers must either leave the code-switched content unanalyzed or treat it as words of the study language. Either of these options could create problems for building an analysis using the database.

If researchers want to analyze intrasentential code-switches, one workaround is to enter all code-switched words as allomorphs of a single headword in the study language lexicon, called something like "EnglishWord" or "SpanishWord." Alternatively, if the user plans to study

---

[14] Text size and color issues (https://groups.google.com/g/flex-list/c/Na1Pqc17wXk/m/y7Tezq4MBAAJ, https://groups.google.com/g/flex-list/c/m5mue4bfiRA/m/8I2tlBK6BAAJ), including their impact on displaying the database to older speakers (https://groups.google.com/g/flex-list/c/l9XTMNL2EVs/m/UZtjiyyZAQAJ), have also been discussed on the FLEx Google Group.

[15] SIL does offer SooSL, a dedicated software program for assembling signed-language dictionaries (https://soosl.net/), but this consists only of the dictionary component. It does not include text collections or a parser.

[16] Lines, termed "phrases" on the FLEx backend, often correspond to utterances.

syntactic characteristics of code-switching, they can create unique "EnglishNoun" and "EnglishVerb" (or parallel) headwords. These workarounds allow users to search for code-switches, but they are labor-intensive: users must individually add each new code-switched word encountered in the transcripts as an "allomorph" of the relevant headword.

Another solution is to translate code-switched items back into the other language. This is a popular option for heavily edited texts for endangered language documentation materials, but the translation solution has drawbacks as well. It does not represent what language users actually say, and it divorces the text transcript from the underlying recordings, which causes problems if the text is later used in forced alignment or other work that requires an exact match between the audio/video and transcript. See Holden 2013:25–27; Mosel 2015:324–325; Dobrin 2021:44–49 for further discussion.

Last, if a text contains code-switches between lines, those can be analyzed. But since FLEx does not come with any lexica, the user will need to manually add lexical entries for every item in the code-switched lines – even if the code-switched content is in English, Spanish, or another high-resource language where lexicons are widely available.

2.2.3 Outcomes

With these assumptions, FLEx encourages the user either to rely only on "mono" data, or to "clean up" more complex text into mono form, following much the same research practices that Boas and the early Americanists used more than a century ago (Epps, Webster, and Woodbury 2017:45). While research based on mono materials has a long and important history in language documentation, it is not the only way to document a language, and it is not sufficient for many research questions. The field needs tools that are capable of analyzing topics – for example, signed languages or multilingualism – that contemporary researchers and revitalization practitioners have highlighted as documentation priorities.

2.3 Collaboration and integration

Besides being built for a specific type of language data, FLEx is also built for a specific type of user – either a missionary with some training in linguistics, or an academic linguist, who works by themselves on a single computer. As a result, FLEx is difficult to use for collaboration. Unlike much professional software used today, the app does not allow multiple users to edit a database simultaneously (another "mono" feature). While users can share databases with one another, the version control process is opaque and lexical entries do not include version histories. Furthermore, collaboration requires either exchanging database backups or accessing the database via an account on LanguageDepot and/or LanguageForge, which must be manually created by SIL. As a result, in practice, only one researcher (or one machine) is able to work in the database at a time, unless they are working on non-overlapping areas.

The lack of explicit versioning within FLEx and its arbitrary reversion of dictionary records makes reconciling joint edits time-consuming and error-prone. For example, if two users edit the same dictionary entry, at sync the conflict is resolved by keeping only one set of changes. The program generates a warning, but the warning does not say which changes have been kept, leaving the user to deduce what the additional changes might have been.[17] It is also not consistent which edits are taken as primary at sync. In our experience, these constraints often mean that research teams spend a great deal of time reconciling edits, developing alternative forms of version and data control, waiting to work until others are finished, or coordinating handoffs of FLEx-specific hardware.

Additionally, the package's collaboration requirements run counter to principles of Indigenous data sovereignty, because the LanguageDepot collaboration interface requires users to upload their data to a server controlled by SIL. As the members of the Indigenous Protocol and AI Working Group (2020:22) write, "Indigenous communities must control how their data is solicited, collected, analysed and operationalized. They decide when to protect it and when to share it, where the cultural and intellectual property rights reside and to whom those rights adhere." Like many terms of use for commercial products (Holton, Leonard, and Pulsifer 2022:56), the LanguageDepot terms of use are fundamentally inconsistent with this level of user control, for example because they grant SIL a license to all uploaded materials.[18]

2.4 Language reclamation

As this data sovereignty discussion suggests, FLEx has many attributes which limit its ability to support language reclamation programs. The remainder of this section focuses on these issues. Our positionality as community outsider language researchers means we have limited experience in discussing the use of FLEx in language reclamation. Thus, we draw on our experience in community collaborations, as well as from the literature on reclamation methods by Indigenous researchers in discussing this topic.

While FLEx is mostly used in language documentation and Bible translation projects, it is not used only in academic settings. Language reclamation programs often make use of materials collected by community outsiders (see e.g. Warner, Luna, and Butler 2007; Baldwin, Costa, and Troy 2016; Lukaniec 2022) and need some way of organizing such materials in order to analyze, use, and share them. FLEx has the potential to be helpful in this process. However, one clear problem for using FLEx in language reclamation work is that the software assumes that language can be studied independently of its social context. For example, as discussed by Lukaniec (2022:321), the dictionary module does not allow users to provide different source (e.g. dialect,

[17] There is an option to compare versions on the LanguageDepot site, but no option for a user to restore from intermediate points. Data loss due to sync issues is discussed in many threads on the FLEx Users Google Group (https://groups.google.com/g/flex-list/c/kxb6Yy93DFM/m/Hs1UzrpzBQAJ, https://groups.google.com/g/flex-list/c/oCGBZJYIlbM/m/qao2TWCzAQAJ).
[18] https://software.sil.org/language-software-terms-of-use/#your-content

register, or speaker) information for different forms of a headword (see §3.3.2 for similar problems in the text module). This assumption is not necessarily surprising – after all, the field of linguistics is built on the treatment of language as an independent, modular system. But language records do have extensive connections outside of language. They record information about material culture, philosophy, and history; encode cultural knowledge; are records of interactions; and can also be important mementos of individuals. As the Indigenous Protocol and Artificial Intelligence Working Group (2020) put it, "Indigenous knowledge is often relational knowledge," so Indigenous projects that center language need to also be able to retain the many relations that language creates. As we discuss further in §3, the data structures of FLEx provide minimal support for describing these relations of the language materials.

Many Indigenous language reclamation programs center the relationship between language and wellbeing in their work. That is, often the goal of a language program is not simply to increase fluency in or prominence for the *language*, but to situate it in a larger network of health, culture, and Indigenous praxis (e.g. McCarty et al. 2018; Leonard 2017; Chiblow and Meighan 2022; Whalen et al. 2022). Indigenous language reclamation programs also place much importance on methodology and collaboration while working on language data. The processes of bringing the language home, connecting with elders and ancestors, and rehousing and resituating the knowledge within the language materials are deeply emotional and challenging – though often also ultimately restorative. That is, language reclamation is not simply about adding words to a database. But software that makes collaboration difficult (§2.3) also makes reclamation difficult.

Whether they work from legacy materials and databases or from contemporary speakers' knowledge, reclamation programs also typically emphasize creating materials that can be circulated in and used by the Indigenous community to promote language transmission, or to repatriate/rematriate cultural and linguistic knowledge. Because such materials are "for the community," it is important to be able to move materials into and out of the database in a variety of formats. As we discuss in §3.2.2-3.2.3 below, however, this import and export process is difficult to achieve in FLEx without extensive workarounds or reformatting.

Thus, the way FLEx is designed facilitates a particular view of linguistic research, while making other types of research more difficult. We offer several concrete examples of this effect in §3. One might say then that such research projects should just use another tool. However, most linguistic software tools have one or more of these problems. Further, as noted above, FLEx is just about the only option if parsed output is needed. And at this point, since so much language documentation material is in FLEx databases, projects may use FLEx because earlier materials used it.

3 Data structures and their effects on analysis

§2 sets up the key ideas around how software design impacts research. In this section, we describe more specific examples of how FLEx data structures and software interact with analysis workflows. These cases were chosen to illustrate the breadth of issues and the importance of this topic for language documentation.

3.1 Morphological parsing

In this first case study, we consider the use of FLEx for parsing morphology. Our goal in this section is to describe the impact of FLEx's assumptions about *grammar*, rather than its assumptions about the user or data structure, which are described in later use cases (§3.2-3.4).

By default, FLEx uses a string-based parser called XAmple, which draws on the lexicon and surface forms of the analyses that the user has approved to recognize patterns and suggest future analyses (Black, Simons, and Zook 2007). This parser functions somewhat like a field notebook for language description. With XAmple, FLEx can be used to store morphological information as the researcher learns about the language. n turn, this information trains the parser to suggest previously-approved analyses for unanalyzed items, as long as these unanalyzed items are orthographically identical to ones that have already been approved. FLEx also has a "Hermit Crab" parser, which implements word-by-word parsing based on user-specified phonological rules. There is some support for specifying word structure to reduce the number of parses – for example, by prioritizing prefixation or suffixation in parsing, or limiting affixes to certain word classes.

3.1.1 Workarounds for parsing nonconcatenative morphology

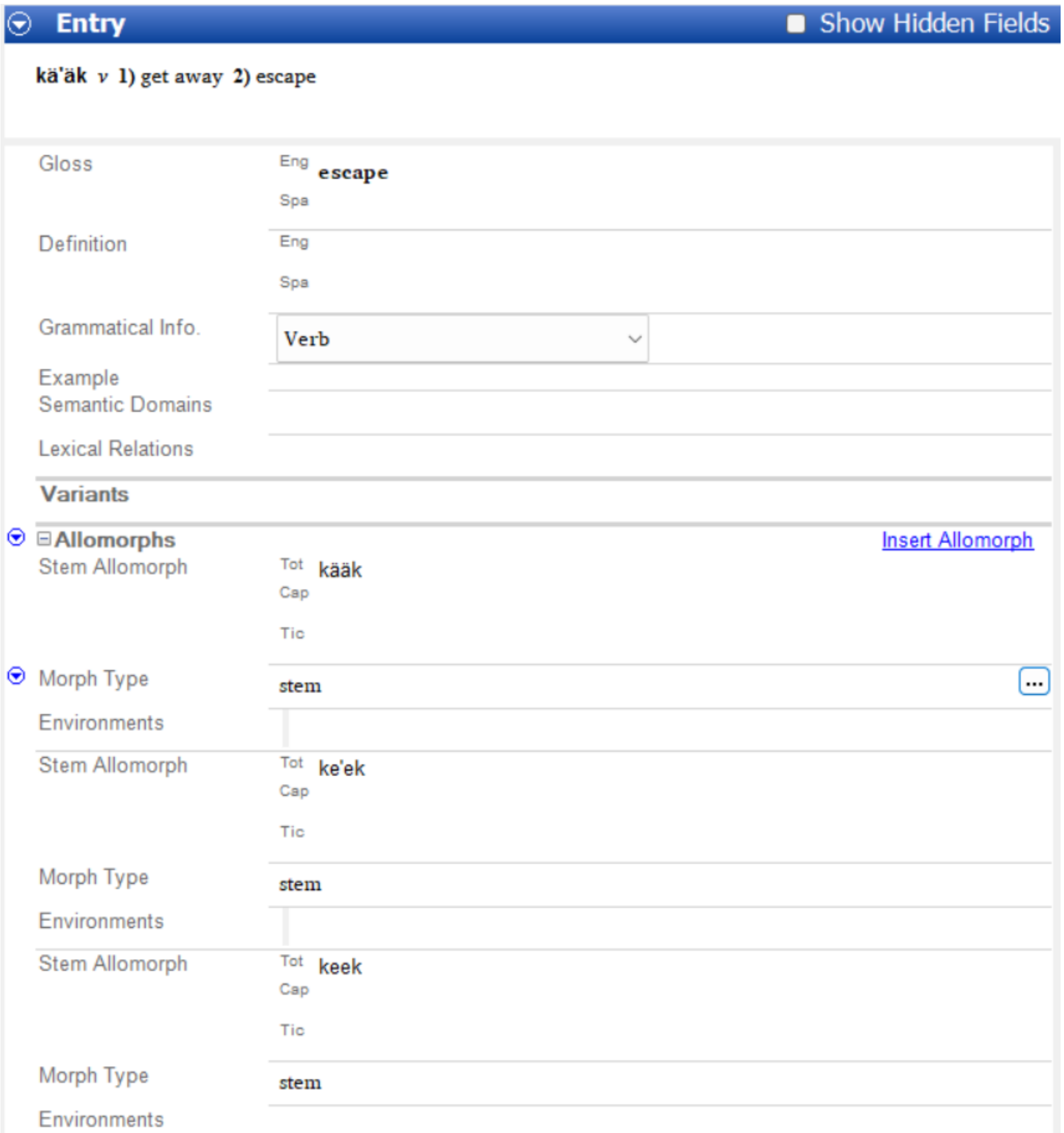
The Xample and Hermit Crab approaches work well for agglutinative morphology or for portmanteau forms where variants can (and should) be exhaustively listed in the lexicon. But for other kinds of morphology, they quickly create problems. Consider a case in which a researcher is describing a language with stem alternations, such as Ayöök (also known as Totontepec Mixe) (Guzmán Guzmán 2012; Suslak 2005). In this language, verb roots tend to consist of CVC syllables. Vowel nucleus shapes and vowel qualities both alternate, such that the same stem can appear with either of two vowel nucleus shapes and either of two vowel qualities. The alternations are crossed, meaning that most roots have at least four distinct forms (Table 1).

Table 1. Ayöök root alternation patterns

Root: **kä'äk 'escape'**	Nucleus Shape A	Nucleus Shape B
Vowel Quality A	kä'äk	kääk
Vowel Quality B	ke'ek	keek

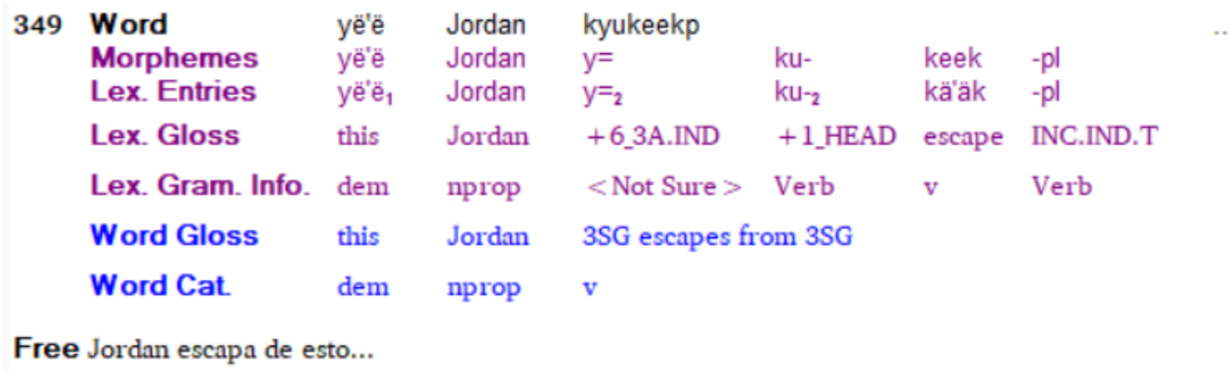
One option is to add all variants to the lexicon, as shown in Figure 1.

Figure 1. Representing the stem alternation of the Ayöök verb kä'äk 'escape' in the FLEx lexicon by creating allomorphs



However, if the user chooses the allomorph solution in Figure 1, no information about the nonconcatenative process will be included in the gloss (Figure 2). This makes it impossible for the user to search for all tokens of verbs displaying the CeeC stem alternant, in order to identify the conditioning environment of this allomorph. Regular and irregular allomorphs are also treated identically.

Figure 2. Analyzing an Ayöök text using stem allomorphs to represent a nonconcatenative process

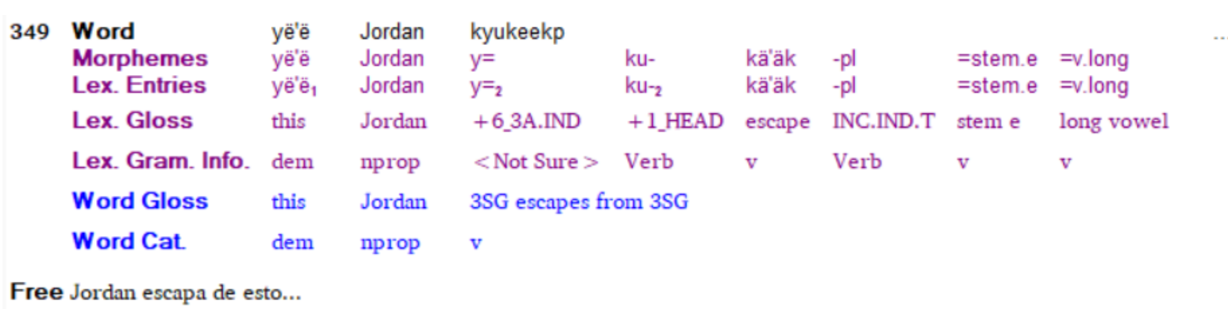


An alternative is to pretend that the process is concatenative and to define a morpheme for each stem type. In this option, each of the possible nucleus shapes and vowel qualities is assigned its own lexical entry. Table 2 provides a list of what the lexical entries for placeholder entries might look like. The analyst can then insert the placeholder morphemes in the gloss line for each verb token, as shown in Figure 3.

Table 2. Placeholder morphemes for tracking an Ayöök stem alternation in FLEx

Vowel nucleus shape or quality	Placeholder "morpheme"
/a/	=stem.ä
/e/	=stem.e
VˀV	=v.glottal
V:V	=v.long

Figure 3. Analyzing an Ayöök text using placeholder morphemes to represent a nonconcatenative process



3.1.2. Implications

Unlike creating allomorphs, creating placeholder morphemes like those in Table 2 will enable the researcher to keep track of all of the relevant stem alternations and their environments in an Ayöök corpus. However, for every token of a lemma where the surface form does not match what is determined to be the headword in the lexicon, the researcher must manually change the surface form to the headword for the program to register the correct lexical item.

Furthermore, this workaround obscures the morphophonological processes at work: claiming that an empty suffix triggers stem alternations just to parse the forms is to make a theoretical (and analytical) claim about morphology simply for the convenience of the software. This misclassification creates additional problems for analysis of concatenative morphology, since the researcher will need to find a way to distinguish between true concatenative morphemes and placeholder morphemes representing nonconcatenative processes.

If the researcher is confident about the conditioning of a nonconcatenative process, they can formulate rules that predict novel analyses using HermitCrab (Maxwell 2003), rather than using the workarounds described above. However, depending on the language's morphology, setting up the appropriate rules may consume even more time than using workarounds.

In sum, the problems here arise from two sources. One is having a single glossing/parsing/lexicography pipeline and maintaining a close relationship between the printed lexicon and the information needed for parsing. The forms that one might want to add to a dictionary for language users are not necessarily the same as the ones that are needed for morphosyntactic glossing. The other problem is that semi-supervised parsing *as a discovery procedure* for linguistic structure requires phonological, morphological, semantic, and syntactic information. Because FLEx's parsers treat the problem as exclusively word-based (and primarily phonological), this cannot be achieved within the software. One solution may be to implement a tagging and dependency parsing approach, which makes use of the lexicon but also has options for specifying parses of larger structures. Another solution would be to integrate FLEx more closely with crosslinguistic NLP projects, such as Unimorph.[19]

3.2 Dictionary and text preparation

In this use case, we examine the advantages and disadvantages of FLEx for preparing dictionaries and text collections, two common types of language documentation publication. Researchers often use FLEx to create a lexicon and collection of texts, then wish to share these materials with others. For work with endangered language communities, the materials should be bi- or multi-lingual and usable for learners of the language of study (that is, include encoding as well as decoding dictionaries).

[19] https://unimorph.github.io/

FLEx has many advantages for this use case, especially for dictionary creation. The app can house both lexical and textual materials, meaning that the dictionary can be built incrementally from the analyzed texts. The lexicon module also allows the user to link lexical entries to examples from the text collection, ensuring that example sentences are culturally relevant and representative of the language as used in the texts.

Within the lexicon module, users can add custom fields in addition to the fields provided by FLEx, and they can define custom categories in many built-in fields (e.g., custom parts of speech). This allows users to adapt the lexicon structure to the grammatical categories relevant to the language and to include cultural and encyclopedic information in the dictionary. Users can also batch edit and search lexical entries and create subsets of the dictionary for different audiences. However, at the point when products are exported from FLEx, serious issues emerge.

3.2.1 Lack of portability and interoperability for dictionary production

Though the FLEx lexicon module is customizable, its dictionary export process is rigid and the export formats are limited. As of the stable release version (9.1.25) from mid-2024, formatted dictionaries can only be exported as XHTML. In theory, since many word processors can read HTML files, this file should be further editable. In practice, however, any changes to the file -- even global changes of font sizes or margins -- result in irreparable formatting errors. Besides XHTML, the other export options are LIFT format (a bespoke XML[20] style for use in mobile phone dictionaries) or Toolbox's standard format lexicon (a text file with each field introduced by a two-letter code, preceded by a backslash).

Once the user has exported a dictionary from the lexicon in HTML, they cannot reorganize entries for the print output. They also cannot add supplemental material to the dictionary, except by manually inserting pages into a PDF printed from the XHTML and viewed in a web browser. Similarly, users have little control over the design or typesetting of the dictionary export: for example, they cannot add illustrations to specific lines of a dictionary entry. FLEx does allow customization of character and paragraph styles, but in our experience, they interact in unpredictable ways. Dictionary formatting and image additions must happen within FLEx, before export, and are therefore limited by what is possible in the program. As of mid-2024 we attempted producing editable exported formatted text, but were unable to do so. The "easiest" way to produce such a dictionary was to export as SFM (text-based markup), export through Toolbox's SFM > Word converter, and then style the character text manually after upgrading the Word 2003 output to a recent word version. Toolbox, however, has not had a major update since 2012, and Microsoft has several times in the past removed access to documents created with outdated versions. This is a fragile workflow.

[20] XML stands for Extensible Markup Language; see Ray (2003) for an introduction to the standard.

By default FLEx lexicons are “decoding” dictionaries: they are designed for users who wish to look up words in the target language and translate them to a language of wider communication, such as English. To go in the opposite direction, users look up a word in the “reversal” (a wordlist of other-language → target language forms). For language reclamation, however, users also need an encoding dictionary: they need to be able to look up words in the language they are fluent in (e.g. English) and find translation equivalents in the target language. FLEx makes it difficult to produce detailed encoding dictionaries. This means that dictionaries produced for learners make them do more work: look up the language form in the reversal list, and then the entry in the main dictionary.

3.2.2 Lack of portability and interoperability for text production

FLEx’s export of interlinearized texts is similarly rigid. When the user exports an interlinearized text from the FLEx texts module, they cannot alter any line breaks, or introduce new punctuation (such as question marks) that is interpreted as line-breaking. FLEx also does not support copying and pasting the baseline text. This removes the interlinearization, making it difficult to insert individual sentences into secondary documents, such as when quoting examples in a reference grammar. Additionally, the processes for exporting textual materials to other file formats are limited, and many do not preserve the very information that makes FLEx a logical choice for such projects. For example, interlinear glossed text must be either copied-and-pasted using a fixed width font, or exported only to SIL’s own text editor. Otherwise, the interlinear spacing is lost.[21]

Some of these issues can be resolved using technical workarounds. Beier and Michael (2022), for example, describe a workflow that allows them to generate a customized dictionary, with front matter, from a FLEx database. Their workflow, however, relies on custom Python and LaTeX scripts in addition to FLEx. Few language documentation teams have the programming expertise necessary to write these add-ons. Moreover, even for teams with a programmer, the FLEx backend uses an extremely idiosyncratic XML structure (cf. the discussion of export formats in §3.3.6), creating a significant learning curve to writing scripts that interface with the program. For text collections, export pipelines such as HTML > PDF > Word keep the text, but not the spacing. In such cases, the interlinear spacing has to be recreated through borderless tables or regular tab placement. Thus, in summary, the options are either to change other parts of the workflow to accommodate FLEx, or to recreate interlinear alignment.[22]

---

[21] In theory, it is possible to export from the export menu to an XML format that is readable in Microsoft Word. However, in our experience, this either produces a corrupt or unopenable file, or a file that treats every line of interlinear text as an equation. In one export, the file resulted in tables where the cell width was the width of the page, so every sentence had a single morpheme per line.
[22] LaTeX users will point out that a FLEx > LaTeX pipeline has no need to retain tab spacing, but this brings additional issues. Not least, LaTeX is hardly used outside of academia, restricting who can edit materials.

Particularly for Indigenous or endangered languages, it is important to be able to use language material in a variety of formats and for a variety of purposes with minimal reformatting (Bird and Simons 2003). These difficulties of importing and exporting language data limit FLEx's utility. The point of using FLEx to parse sentences is negated when parsed data cannot be easily exported without extensive reformatting.

3.2.3 Lack of portability of the database file

Besides these issues creating publication-ready documents, it is difficult to archive copies of FLEx databases securely. If researchers want to deposit their FLEx database in an archive, they will need to decide between archiving individual XML files in FLExText or XLingPaper format, or archiving a backup of the entire database, with the extension .fwbackup. The XML files are more human-readable and much more obviously related to the texts, but they do not include the project lexicon and therefore <u>cannot</u> be used to restore the database. FLExText exports also do not include speaker information, or any information added to a "notebook" record in the text information window. Restoration is possible only with the entire .fwbackup file, making it very easy for users to unwittingly archive incomplete materials. The user interface does not make this difference between XML and .fwbackup files clear, further increasing the risk of data loss during archiving. This also makes it difficult to archive subparts of the project sustainably.

3.2.4 Implications

The comprehensive, customizable lexicography tools offered by FLEx encourage the user to invest a great deal of time in collecting language materials and editing a lexicon. Though the text module is less well-designed - as discussed in our next case study - it also allows users to invest a great deal of effort in collecting and analyzing texts. But at the point of publication, the difficulty in editing the output of the dictionary and text exports discourages users from investing time in customizing the dictionary or text collection as a product. This has been a problem since the first release of FLEx (cf. Butler and van Volkinburg 2007).

To summarize, the dictionary export process leads to a mismatch between the content of FLEx-based publications, which may be excellent and highly adapted to the language and culture, and their appearance, which by default will be awkwardly laid out and uncustomized. This unfinished appearance makes publications less likely to be used. It also sends a negative message about the importance of the publication - and the effort of the research team - to language community members and other end-users. Finally, the poorly designed user interface for exporting to non-publication-ready formats, such as FLExText, raises serious risks of data loss when archiving a FLEx database. Table 3 summarizes these implications of FLEx for dictionary and text preparation.

Table 3. Implications of using FLEx for preparation of dictionaries and text collections

Attribute	Encourages	Discourages
Putatively comprehensive lexicography tool	Using FLEx as the central locus of lexicographical work	Branching ideas for other kinds of lexicographical work
Limited but straightforward export options for texts and dictionaries	Shipping off texts and dictionaries immediately after their analysis/curation	Detailed typesetting configurations (all options are formats that are difficult to modify out of the box) Respect for/Credence to the care put in to curate these resources
Export destroys interlinear alignments	Disseminating only baseline text	Exporting text collections for sharing
Texts can be backed up and exported in human-readable formats (XLingPaper, FLExText)	Archiving individual text files (do not allow restoration of lexical database)	Archiving entire .fwbackup file (only format allowing restoration of entire database)

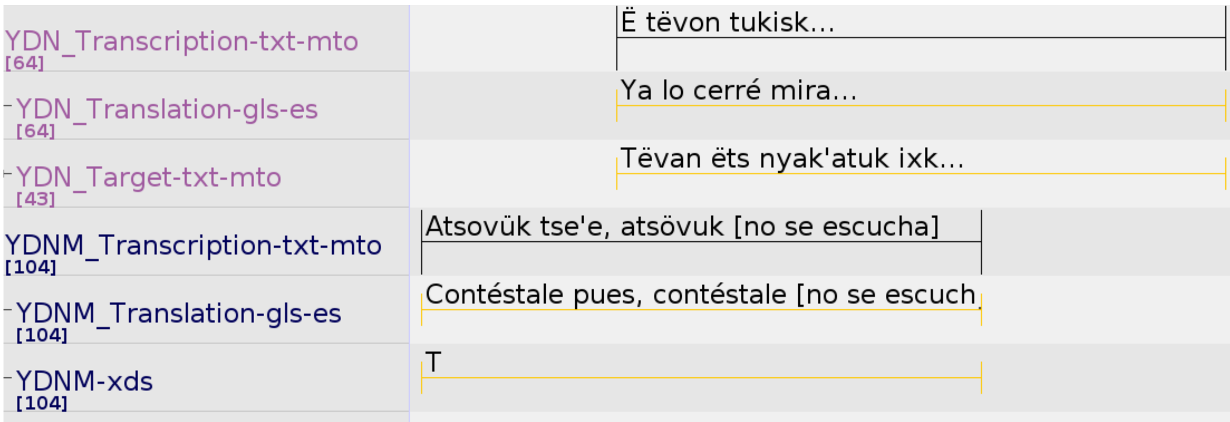
3.3 Transcript analysis

In this third use case, we explore the use of FLEx for research on child language. Since many child language studies are based on recordings of child-caregiver interaction, most of our points also apply to research on conversation between adults. Child-language research data brings up many points that make digital data infrastructure concerns clear, including interoperability, customizability of fields, and customizability of data structures. That is, this use case crystallizes a number of issues that also appear in other subfields.

Consider the situation where a team of child language researchers transcribes and analyzes recordings of child-caregiver interactions. Using ELAN (Wittenburg et al. 2006) or CLAN (MacWhinney 2000) is currently standard for annotating such collections.

These recordings definitionally include at least two participants: the primary child and their caregiver. Many also include other participants, such as siblings or other caregivers and adults. When annotating interactions, researchers attribute turns in ELAN or CLAN to speakers and often also label them for addressee type, as shown in Figure 4 (e.g., the “T” code in the “xds” tier in Figure 4 denotes a turn addressed to the primary child). The speaker information allows users to distinguish the primary child’s speech from other participants’, while the addressee information allows them to distinguish speech directed to children vs. to adults. Additionally, child language data always includes some child utterances that are ungrammatical in the adult grammar. This also impacts the ELAN/CLAN annotation structure: ungrammatical turns are labeled as ungrammatical and annotated with the equivalent adult form, as shown in the 'YDN_Target-mto-txt' tier in Figure 4.[23]

Figure 4. A portion of an ELAN (.eaf) transcript of child-caregiver interaction from the Ayöök corpus. Each turn is annotated for speaker and, in some cases, addressee; ungrammatical turns are annotated with their grammatical ("target") equivalent



As well as these annotation types, child language researchers often need to add morphological analyses to their transcripts. These analyses allow researchers to explore theoretical questions - for example, whether adults use less complex morphology while speaking to children - as well as curriculum-oriented and applied issues, such as identifying the nouns which children hear and use most frequently. Materials of this type can also feed into language documentation programs, where they may be combined with work on adult language use (Hellwig and Jung 2020), or in crosslinguistic projects, where the interlinear glossing is crucial. Any publications which include transcript data will also require interlinearization.

For this use case, the greatest advantage of FLEx is its powerful parser. With large-scale collection of data on language use in interaction, researchers will likely need to analyze tens of thousands of turns at talk. This is time-intensive in any scenario, but can be done several times faster using the FLEx parsers. If the user has a digital lexicon of the language, they can speed up the process further by importing the lexicon to FLEx before analyzing the transcripts, which reduces the number of lexical entries they need to create during analysis. Furthermore, since their analyses will be linked to entries in the FLEx lexicon, they will have protection against glossing inconsistencies caused by typos or other human errors. However, alongside these advantages, the researchers will encounter an enormous number of problems with analyzing interactive transcripts in the text module.

[23] Ungrammatical utterances are also frequently part of elicited adult language documentation corpora.

3.3.1 Interoperability: cumbersome import process

FLEx and ELAN have limited interoperability unless users adapt them substantially, limit interchange between files to a single one-way[24] import, or use a custom interchange format. Child language researchers encounter problems with FLEx when they import ELAN transcripts, during analysis, and again at export. To use the app's native import process, users need to alter their ELAN transcripts by (1) renaming types according to FLEx's required format; (2) renaming tiers; and (3) creating additional tiers that have no purpose outside of the FLEx import (e.g., a tier giving a text title). FLEx offers no built-in way to automate these time-consuming changes or to batch import ELAN files. There is also no built-in CLAN import.

3.3.2 Suppression of participant information

When users view imported ELAN transcripts in the Texts & Words module of FLEx, they cannot see the speaker and addressee codes from the ELAN file, as shown in the screenshot in Figure 5. Speaker information is imported from ELAN on the FLExText backend, but FLEx does not display it to the user. Furthermore, addressee information is not imported at all. Suppressing this information makes conversation transcripts confusing and difficult to analyze, since users cannot tell if adjacent turns come from the same speaker or different speakers. More important, suppressing participant attributions de-contextualizes language from speakers (§2.4) and fails to recognize speakers' intellectual property rights over their language use (§2.3).

Suppressing participant information also makes the FLEx search feature almost completely unusable for many common child language research questions. For example, users cannot search only on child speech, or only on child-directed speech. To work around the suppression of participant information, researchers can add a speaker attribution, such as "Child1", or "Caregiver", to the beginning of each annotation in ELAN and then use the native import function of FLEx. The speaker attributions will then appear as part of the text baseline. This workaround is unsatisfactory: it adds even more time to the import process, and still does not allow users to view addressee information during analysis.

[24] That is, if a user transcribes audio and/or video in ELAN, then imports those transcriptions for analysis into FLEx (without ever re-exporting), the procedures are straightforward. This workflow resembles workflows from the era of analog tape recordings, where a tape would be transcribed but subsequent analysis would usually be done from the written text (without reference to the tape). This is another example where our default workflows reflect earlier uses of technology, where linking text to transcript was impossible.

Figure 5. A transcript from the Ayöök child language database, as viewed in FLEx. All participant, addressee, and timestamp information has been removed

An alternative is to import ELAN participant information as a note or (as of FLEx 9.1) a custom field on each line of the FLExText. The advantage of this strategy is that, because lines can have an unlimited number of notes, users can view both speaker and addressee information as separate notes. The disadvantage of using notes is that it requires circumventing the native import process, meaning that users must have the programming skills to create a custom ELAN-FLEx interchange format. Custom fields offer a lower-effort solution, but only for Windows users, since FLEx 9.1 is not currently available for Linux.

3.3.3 Low search performance

Even when researchers successfully use workarounds to add participant information, in practice they still cannot search only child or only adult turns in FLEx, because of the poor performance of the FLEx search features. Standard searches without regular expressions (regex) work very well - they are easy to execute from a lexical item or in the search function. But simple regex queries – for example, a regex search for the string "Child" followed by a specific morpheme – frequently crash the program or machine. Complex concordance searches cause the same problems. These search performance issues have existed since the first version of FLEx (Butler and van Volkinburg 2007; Rogers 2010).

Furthermore, none of the FLEx search or concordance functions allow users to save parsed results to an external file. As a result, performing statistical analyses of FLEX corpus material requires many additional steps – exporting each text as a FLExText, then exporting the FLExTexts to ELAN, and finally exporting the ELAN file to a CSV which can be read into a statistical software package. This raises significant barriers to quantitative research on acquisition, sociolinguistic variation, and many other topics in Indigenous languages.

3.3.4 Inability to pair actual and target turns by children

If researchers enter both children's actual (potentially ungrammatical) turns, as well as their associated target (grammatical) turns, as time-aligned tiers in ELAN, then during analysis in FLEx, they will be able to view them as separate lines in the baseline. But because FLEx does not allow multiple surface-level representations of the same turn, the import will break the association between the actual and target turns. This means that researchers cannot search the FLEx database for any specific target turn-actual turn mapping – that is, they will not be able to search for any particular error type, such as omission of agreement, or overgeneralization.

As a solution to this issue, FLEx developer Beth Bryson suggests to us that researchers could enter target and actual forms as the same baseline line, using different writing systems for each form. This workaround is infeasible because it requires a one-to-one-mapping between words in the target and actual turns, which generally does not exist in child language. Even if it were feasible, the workaround also makes an implicit and unsupported claim (cf. §3.1) that child language is an alternative representation of adult language, rather than a stage of development.

In our own research, we have solved the target-turns issue by circumventing the native import/export process of FLEx, so that target and actual turns are imported as separate baseline lines but associated via a unique note [citation redacted for anonymous review]. But because of the poor performance of FLEx search, this still does not allow us to search only actual or only target lines in FLEx, or to systematically compare actual and target turns. Instead, we export the transcripts in FLExText XML format and read in the XML to other apps for analysis.

3.3.5 Inability to flag ungrammatical turns

An issue affecting both child language research and research using grammaticality judgments is the treatment of ungrammatical turns. In the text module of FLEx, some users need to include ungrammatical turns in the Baseline transcription so that they can be morphologically analyzed and searched. This might be necessary, for example, in order to analyze patterns of child errors or to search elicited data for judgments against specific morpheme combinations. However, the text module does not include an option to mark lines as ungrammatical. Of course, researchers can use a symbol like '*' to indicate ungrammatical turns in the text baseline, but the standard search features do not allow users to filter results based on grammaticality symbols. To search only grammatical or ungrammatical turns, researchers will again need to use the underperforming regex search.

3.3.6 Interoperability: Problematic export to ELAN

Given the poor performance of the FLEx search features, child language researchers need to export FLEx texts into another format for search when their analysis is complete. Exporting back to ELAN is ideal, since it allows users to access speaker and addressee information again, as well as having parsed text synced to the original audio and/or video. But in practice, the built-in FLEx-to-ELAN export process is riddled with bugs.

In ELAN files exported from FLEx, phrase-level transcriptions are erased and replaced by empty phrase-level annotations dominating a tokenized word tier, and speaker attributions in tier names are replaced by arbitrary letters. For example, compare Figure 6, which shows the original ELAN transcription of a conversation in Ticuna, to Figure 7, which shows the same conversational sequence when analyzed in FLEx and exported back to ELAN. Figure 6, the original, displays the entire Ticuna turn by speaker RGW (her initials) as a single annotation, and it attributes the turn to her in the tier name. But after the transcript is analyzed in FLEx and exported back to ELAN, as shown in Figure 7, the turn is no longer represented as a single annotation. The 'A-phrase_segnum' tier includes an annotation with the same timepoints as the original turn, but it contains a meaningless number in lieu of the transcription, and the individual words are transcribed only on a tokenized word tier. Furthermore, RGW's speaker attribution in the tier name is replaced by the arbitrary letter "A" (even though the tier still has the participant attribute "RGW" in the XML). These changes push the speakers' identities away from the main transcript, decrease the clarity of the transcript to users, and operationally, make it much harder to search for multiword strings.

Figure 6. Original ELAN transcription of the first turn in a Ticuna conversation (California Language Archive reference [2018-19.024](), file: tca_20170527_disc_002.wav, 0:45)

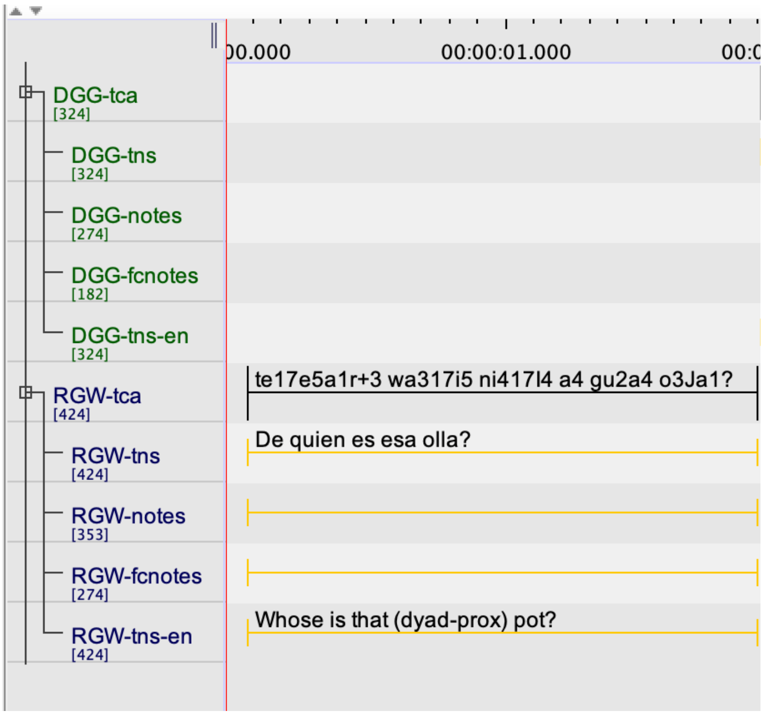
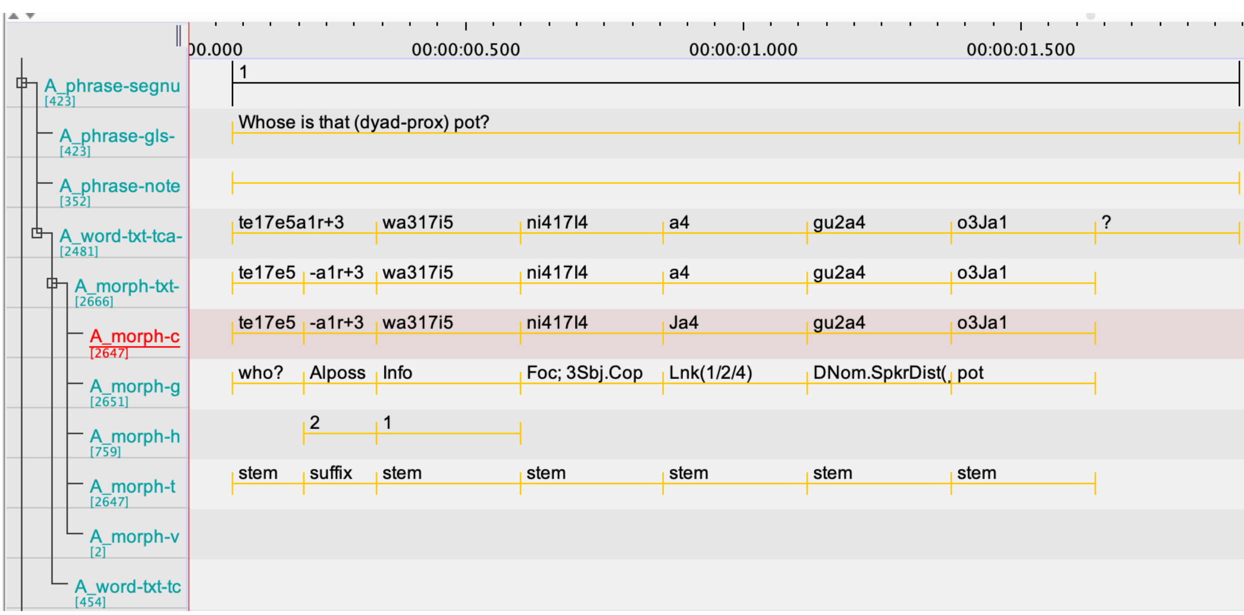


Figure 7. Result when the Ticuna transcript shown in Figure 6 is analyzed in FLEx, then exported back to ELAN



The loss of phrase-level transcription shown in Figure 6 and Figure 7 cannot be resolved by exporting to any alternative format. Phrase-level transcriptions are absent in the FLExText backend and therefore also absent in the app's other export formats, such as FLExText XML and

XLingPaper. Researchers can work around this issue only if they have the technical skills necessary to edit the XML export by concatenating the text of the word nodes and copying it to the phrase nodes. Similarly, FLEx does not offer text export to flat formats such as CSV. Users who need CSVs for statistical analyses must export to one of the XML formats offered by FLEx, and then parse the XML in another app.

3.3.7 Implications

Through these attributes, FLEx encourages certain types of research and discourages others. These are summarized in Table 4.

Table 4. Implications of FLEx data structures for transcript analysis

Attribute	Encourages	Discourages
Import of time-aligned files from ELAN is cumbersome	Analyzing only born-text materials	Analyzing audio or video transcripts
Text module suppresses speaker and addressee information	Analyzing only monologues	Studying conversation Comparing speech by different types of speakers, e.g., learners vs. fluent speakers Comparing speech addressed to different types of listeners, e.g., to children vs. to adults
Regex searches and complex concordance feature crash machine or program	Using simple concordance as only search tool	Using all available search features
Search results cannot be exported to CSV	Manually entering search results into other software for analysis	Creating fully reproducible, dynamically updating analysis pipeline (à la R Markdown)
ELAN export is poorly structured	Using FLEx as endpoint of analysis	Re-exporting to ELAN after analysis
Exports are in formats not commonly used by linguists or communities, e.g., no flat formats, no phrase-level transcription	Using FLEx as endpoint of analysis	Using FLEx as one element of pipeline for later analyses with, e.g., R

3.4 Historical text analysis

Previous sections have focused on language data gathered directly with research participants. Language documentation projects also involve work with written materials collected by other linguists. In this section, we focus on issues related to work which involves analyzing and interpreting older sources (from the early twentieth century and before) for reclamation, historical analysis or other research. Recent examples include Begay (2017) for Wailaki and Brixey and Artstein (2021) for Choctaw. Lukaniec (2022) gives an overview of these processes with examples from Wendat.

Since the primary aim of such a project is the transcription, translation, and analysis of textual material, this type of work should be straightforward in a program like FLEx. However, even fairly simple historical text collections raise problems for FLEx's data structures, and the requirements of linguistic work – particularly morphological parsing – mean that software developed for other historical or annotative work is inappropriate.[25] The problem can be summarized as follows: working with historical texts involves intertwined presentation and analysis of primary data. But FLEx data structures require making decisions in advance of an analysis, or (as described in the previous section) using workarounds that create their own problems.

3.4.1 Metadata

Working with textual collections from different sources and periods introduces additional required metadata. In addition to the contributor, situational, and item format metadata that Sullivant (2020) discusses, historical documents often have format metadata, such as page layout information.[26] Contributor metadata is particularly important for studying variation among speakers and for properly attributing individual or family intellectual property (cf. Lukaniec 2022). Such metadata may need to be associated with items at the word level, as well as to a text as a whole. However, FLEx does not support metadata at levels other than the entire text (as discussed above for conversational data).

Further, for the metadata that is supported, FLEx's search functions also limit historical text use. While it is possible to specify a speaker, genre, or date in the metadata of a text, genre information is not exported in the FLExText output, and text dates cannot be used to sort or filter. In other words, it is possible to enter metadata about the text, but impossible to use that information when searching or exporting the corpus (cf. §3.3.2). The ability to sort and search by metadata categories is particularly important for text collections that span centuries.

[25] We have in mind software such as Mirador (for image collation and annotation) or NVivo (for document annotation and knowledge graph creation).

[26] Paradigms, for example, contain information about word relationships that are represented spatially. Anderson (2020) gives examples from paradigms in a Tunica manuscript.

3.4.2 Word-Level Annotation

Historical projects typically have many interleaved and complex annotations in addition to interlinear glossing. For the Laves Bardi text collection, for example, Bowern (2003) includes Laves' original annotations, comments on Laves' texts and translations, comments on his annotations, and meta-commentary on text subjects. That is, there is information provided by Laves, information provided by Bardi elders on the original speakers' materials and Laves' comments, and Bowern's annotations that link to other materials in the collection and clarify grammatical and lexical concepts. These are annotations at the word, phrase, sentence, and annotation block level. However, none of these levels, except the sentence or line level, allow annotation in FLEx.

3.4.3 Multi-Content

Historical text collections also usually contain materials from different periods, authors, and transcribers. The source materials often use varied transcription conventions, which may be poorly documented. Researchers usually want to keep track of these conventions and not standardize them immediately, as the process of conducting research with the text collection may change their knowledge of how the conventions work.[27] Some of these conventions might be "surface" (that is, simply convertible, such as using <ṅ> or <ng> for /ŋ/). Others may be more embedded in the media, such as using spatial layout on the page to signal language, as in the Bates collection[28] typescripts.

FLEx handles such varied textual inputs through the twin concepts of "language" and "writing system". That is, a project may include multiple languages (object language, analysis language, and lingua francas, for example) and within a language there may be multiple writing systems (practical orthography, IPA, syllabary, etc.). Some types of material fit easily in this dichotomy. Standard Uzbek Cyrilic vs. Standard Uzbek romanization, for example, has straightforward mappings between character sets. When dealing with multiple different inputs for older materials, however, textual differences might not be purely spelling variants (though they may include spelling variants); they aren't different writing systems (though they may be); and they aren't different languages or varieties (though they may be). They are different doculects (Cysouw and Good 2013) with different conventions of representation, all of which can be mapped onto the standards for the documentation project once they are understood.
Historical text collections also share some of the problems that are discussed in §3.3.2 on child language. The contents of the "text" line in this data are not uniform – need not be grammatical, need not be in the same language, etc. But FLEx assumes that lines are uniform, and this

[27] For example, fieldnotes from archival sources may not be explicit about transcription conventions, and only subsequent research, enabled by the corpus, might lead to an understanding of the conventions used.
[28] http://bates.org.au/text/

assumption – along with the lack of word-level annotation – makes it difficult or impossible to track the type of information that the researcher needs.

3.4.4 Implications

FLEx encourages certain metadata and annotation practices in historical text work and discourages others. Table 5 summarizes these issues.

Table 5. Implications of data structures for historical text work

Attribute	Encourages	Discourages
Must associate all language input with a "Language" and "Writing System"	Irreversible decisions before analysis has been undertaken	Waiting until later in analysis to assign text to language or orthography
Preconfigured, text-level metadata	Structured metadata (an advantage)	Metadata that is needed for the types of texts used
Simple annotation	Minimal annotation, annotation at the word type level	Annotation at level of single word token

4 Discussion: workarounds and alternatives

In this section, we explore how the developers of FLEx and the FLEx user community have responded, so far, to the problems discussed in §2 and §3.

4.1 Attempted solutions by FLEx developers

Over the lifespan of FLEx, the program's developers have tried to remedy some of the most common user complaints. However, these changes still do not meet the needs of many users. For example, the developers expanded the OS compatibility of FLEx to Linux in addition to Windows, but have not expanded to MacOS, despite the high demand for that feature. Similarly, as of FLEx 7.3, users can collaborate on a database with people using other computers, but collaboration still requires a finicky, opaque version control process. Community collaboration is now done primarily through LanguageForge, but this only applies to lexicon editing.

Recent changes to FLEx have also responded to analysis- and publication-related complaints about the program. These changes, however, are also less than satisfactory. For example, previous versions of FLEx assumed that the outputs of the lexicon and corpus would always take the same form. Over the last several versions, the program's developers have acknowledged that users need to create different kinds of resources from FLEx databases, and they have added compatibility with more programs. Yet customization of dictionaries and texts is still extremely limited, and the output file formats still make it difficult for average users to modify either the aesthetics or the content of outputs. In the same vein, as of FLEx 9.1, users can add custom fields to each line of text. Custom fields allow users to organize various forms of line-level annotations that were more difficult to organize in previous versions of FLEx, but they still make the assumption that all annotations should appear at the utterance level, rather than more or less granular levels, such as word or paragraph.

We do not raise these points as naive criticism of software development. We understand that FLEx is a complex program, developed by a small team of dedicated individuals who face complex challenges and who have a view about what linguistics is. But a piece of software that is so tightly integrated into language documentation workflows should not be beyond critical analysis, just as points in linguistic theory are subject to critique and change.

4.2 Attempted solutions by users: A culture of workarounds

Despite the changes to the software outlined above, there is a persistent mismatch between researchers' goals and FLEx's functionality. In response to this, the FLEx user community has developed a pervasive culture of workarounds. Some of these workarounds are small tweaks within the database, such as creating an 'EnglishWord' headword to annotate instances of code-switching. But other workarounds, like building custom converters to move data between FLEx and other apps (Beier and Michael 2022; Skilton et al. in press), require programming knowledge – meaning that teams must either recruit programmers or upskill on their own in order to create working dataflows.

Other workarounds create problems for future analysis. FLEx builds numerous linguistic structures into the database (e.g., roots, stems, clitics, derivational affixes) as preset options. This is intended to speed up analysis. However, because those items are associated with particular parsing behavior, users are encouraged to pick whatever category produces their desired parsing behavior, even if it results in an incorrect or unsupported claim about the grammar. A 2023 post to the FLEx-List Google group, for example, encouraged a software workaround where absolutive case affixes were treated as derivational morphology.[29] Making theoretical claims about morphological structure that are purely driven by software needs is problematic (cf. §3.1.2). An analyst who uses the database in the future may assume that absolutive-as-derivation as a theoretical claim about morpho-syntax, rather than a workaround driven by software.

This need for workarounds, whether small or large, creates serious bottlenecks. At best, it slows the progress of language work. At worst, when teams lack the resources to create a workaround, they may abandon FLEx and either choose not to gloss materials, or switch to an inefficient strategy such as glossing text entirely by hand. Furthermore, as well as being costly in time and money, workarounds are fragile. They have to assume particular XML structures in the input and

[29] https://groups.google.com/g/flex-list/c/TFQ6czlhWzk/m/9aEiAtpwCgAJ

output of FLEx, and this means they are liable to break when the app is updated, or lead to incompatibility between FLEx versions. They also tend to be project-specific, meaning that others cannot use them without specialized knowledge and a time commitment.

4.3 Reimagining language tools

What, then, would alternative software look like? Is there a way to capitalize on the advantages of our current tools while avoiding their pitfalls? Or is language data so inherently complex, and so inherently tied to other aspects of culture, that *any* analysis tool is going to have issues similar to those identified here? A full answer is a book in itself, but several requirements emerge from this analysis.

The first is linkability within a documentation collection. Many linguists use FLEx because of its utilities for storing lexical information and linking it to a parser. Tools are available for training parsers on corpora in supervised and semi-supervised ways (cf. the tools in the Universal Dependencies project; Marneffe et al. 2021), but their results tend not to be easily integrated back into a workflow that includes multi-use parsed corpora. The universal dependencies output, for example, annotates morphological features at the level of a word, but not morphemes within a word. Rather than integrating all the other tools into a parser, perhaps the best solution is to develop portability for parsed data. Some attempts have been made in that direction (such as Kratylos; see Kaufman and Finkel 2018).

Linking and sharing outside the collection is also crucial. For a relational approach to language and for projects where links to external items are crucial, being able to maintain a "knowledge base" that *also* includes accessible language material is vital. Some social science research tools, such as NVivo, already facilitate this type of work. But they typically bring their own assumptions about language and data, as well as lacking the parsing and dictionary options that linguists require.

Perhaps the ultimate goal is not, then, a single tool that does everything that every person working on language needs. That is, the solution is not a single solution, but rather a set of common data formats and a way to render them, along with support and incentives to use them. Some possible premises for data tools include the following: (a) Linguistic data must be linkable and shareable outside a linguistic analysis, because it is inherently related to other tangible and intangible cultural material. (b) Linguistic data raises sovereignty and intellectual property rights that must be recorded and respected. (c) Linguistic data is fragile and must be safeguarded. (d) Linguistic data is variable, and the techniques to analyze and preserve that variability must be part of the toolkit. (e) Linguistic software should *facilitate* insights, not constrain them. (f) Linguistic software should be both backwards- and forwards-compatible; modular tools must be able to pass data back and forth, through a common standard.

5 Conclusions

As a discipline, language documentation relies on old software. Many of these older digital tools are still functional. Though they may not encourage collecting new forms of data, they also do not prevent it – for example, ELAN is not designed for work with children, but its data structures are flexible enough that it is widely used in child language research nevertheless. Other legacy tools, however, have negative impacts on the field because of their monopoly status and rigidity. FLEx is the prime example here: through all of the data and analysis structures that we have described above, the software encourages users to retain historic methods (§2.2) and raises barriers to innovative types of research (§§3.3-3.4).

In sum, we are faced with an infrastructure problem for the discipline. Language documentation workers are traveling the same (or similar) paths and are encountering the same barriers. As we have shown in the previous sections, teams are mostly building their own detours around these barriers – creating workarounds – rather than repairing the path. Going forward, the discipline will need to reflect on the consequences of upholding this status quo. This reflection offers us an opportunity to collectively construct our future relationship with digital infrastructure. To do so, language workers will need a centralized space to discuss technological problems that arise and ideas about potential solutions. Many of today's ad hoc solutions could apply to projects other than the ones that they were created to support. Yet this potential is not often realized, whether because the solutions are idiosyncratic or because they never circulate beyond a single research team.

In contrast to the status quo, language documentation needs a digital infrastructure that is flexible and generalizable enough to accommodate teams' diverse goals, questions, and workflows. This infrastructure also needs to be sustainable, in two ways – it needs institutional support, in the form of funding, updates and technical support, and it needs a user community where people can exchange solutions to common problems, rather than duplicating efforts in isolation. These forms of support are what have made FLEx the standard, but they do not need to be unique to FLEx. Instead, our goal is to begin a conversation about how the field can create digital infrastructure that is shared, supported, and flexible.

6 References

ANDERSON, P.T. 2020. *Revitalization lexicography: the making of the new Tunica dictionary*. Tucson: University of Arizona Press.

BALDWIN, DARYL; DAVID COSTA; and DOUGLAS TROY. 2016. Myaamiaataweenki eekincikoonihkiinki eeyoonki aapisaataweenki: A Miami Language Digital Tool for Language Reclamation. *Language Documentation & Conservation* 10.394–410. doi:http://hdl.handle.net/10125/24713.

BEGAY, KAYLA. 2017. Wailaki grammar. Berkeley: University of California dissertation.

BEIER, CHRISTINE; and LEV MICHAEL. 2022. Managing Lexicography Data: A Practical, Principled Approach Using FLEx (FieldWorks Language Explorer). *The Open Handbook of Linguistic Data Management*, ed. by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 301–314. The MIT Press. doi:10.7551/mitpress/12200.003.0029.

BIRD, STEVEN; and GARY SIMONS. 2003. Seven Dimensions of Portability for Language Documentation and Description. *Language* 79.557–582. doi:10.1353/lan.2003.0149.

BLACK, ANDY; GARY SIMONS; and KEN ZOOK. 2007. *Parser and interlinear text*. https://software.sil.org/fieldworks/wp-content/uploads/sites/38/2016/10/Parser-and-interlinear-text.pdf.

BOERSMA, PAUL; and DAVID WEENINK. 2024. Praat: Doing phonetics by computer. praat.org.

BOWERN, CLAIRE. 2003. Laves' Bardi texts. *Maintaining the links: Language, identity and the land. Proceedings of the Seventh FEL [Foundation for Endangered Languages] Conference*, ed. by Joseph Blythe and R. McKenna Brown, 137–143. Broome: Foundation for Endangered Languages.

BOWERN, CLAIRE. 2015. *Linguistic fieldwork: A practical guide*. Second edition. New York: Palgave Macmillian.

BRIXEY, JACQUELINE; and RON ARTSTEIN. 2021. ChoCo: a multimodal corpus of the Choctaw language. *Language Resources and Evaluation* 55.241–257. doi:10.1007/s10579-020-09494-5.

BUTLER, LYNNIKA; and HEATHER VAN VOLKINBURG. 2007. Review of Fieldworks Language Explorer (FLEx). *Language Documentation & Conservation* 1.100–106. doi:http://hdl.handle.net/10125/1730.

CHELLIAH, SHOBHANA LAKSHMI; and WILLEM JOSEPH DE REUSE. 2011. *Handbook of descriptive linguistic fieldwork*. Dordrecht: Springer.

CHIBLOW, SUSAN; and PAUL J. MEIGHAN. 2022. Language is land, land is language: The importance of Indigenous languages. *Human Geography* 15.206–210. doi:10.1177/19427786211022899.

CYSOUW, MICHAEL; and JEFF GOOD. 2013. Languoid, doculect and glossonym: Formalizing the notion "language." *Language Documentation & Conservation* 7.331–359. doi:http://hdl.handle.net/10125/4606.

DAVIS, JENNY L. 2017. Resisting rhetorics of language endangerment: Reclamation through Indigenous language survivance. *Language Documentation and Description* 14.37–58. doi:10.25894/LDD147.

DOBRIN, LISE. 2021. The Arapesh "suitcase miracle": The interpretive value of reproducible research. *Language Documentation and Description* 21.37–69. doi:10.25894/LDD16.

DOBRIN, LISE; and JEFF GOOD. 2009. Practical Language Development: Whose Mission? *Language* 85.619–629.

DUNHAM, JOEL; GINA COOK; and JOSHUA HORNER. 2014. LingSync & the Online Linguistic Database: New Models for the Collection and Management of Data for Language Communities, Linguists and Language Learners. *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, 24–33. Baltimore, Maryland, USA: Association for Computational Linguistics. doi:10.3115/v1/W14-2204. https://aclanthology.org/W14-2204.

EPPS, PATIENCE L.; ANTHONY K. WEBSTER; and ANTHONY C. WOODBURY. 2017. A Holistic Humanities of Speaking: Franz Boas and the Continuing Centrality of Texts. *International Journal of American Linguistics* 83.41–78. doi:10.1086/689547.

GONZALEZ, J. L.; ANUSCHKA VAN'T HOOFT; JESUS CARRETERO.; and VICTOR J. SOSA-SOSA. 2017. Nenek: a cloud-based collaboration platform for the management of Amerindian language resources. *Language Resources and Evaluation* 51.897–925. doi:10.1007/s10579-016-9361-8.

GRZECH, KAROLINA; and SELENA TISALEMA SHACA. 2022. Building trust on Zoom: A workflow for language documentation via videoconferencing software. *Language Documentation & Conservation* 16.79–97. doi:http://hdl.handle.net/10125/74673.

GUZMÁN GUZMÁN, VERONICA. 2012. Las construcciones aplicativas en el mixe de Totontepec. Mexico City: Centro de Investigaciones y Estudios Superiores en Antropología Social master's thesis.

HARVEY, DAN. 2022. ACORNS: Acquisition of Restored Native Speech. http://cs.sou.edu/~harveyd/acorns/index.php.

HELLWIG, BIRGIT; and DAGMAR JUNG. 2020. Child-directed language – and how it informs the documentation and description of the adult language. *Language Documentation & Conservation* 14.188–214.

HOLDEN, JOSH. 2013. *Benasnì — I remember: Dene Sųłiné oral histories with morphological analysis*. Leiden: Brill.

HOLTON, GARY; WESLEY Y. LEONARD; and PETER L. PULSIFER. 2022. Indigenous Peoples, Ethics, and Linguistic Data. *The Open Handbook of Linguistic Data Management*, ed. by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 49–60. The MIT Press. doi:10.7551/mitpress/12200.003.0008.

INDIGENOUS PROTOCOL AND ARTIFICIAL INTELLIGENCE WORKING GROUP. 2020. Guidelines for Indigenous-centred AI design v. 1. *Indigenous Protocol and Artificial Intelligence Position Paper*, ed. by Jason Edward Lewis, 20–24. Honolulu, Hawai'i: The Initiative for Indigenous Futures and the Canadian Institute for Advanced Research (CIFAR).

KAUFMAN, DANIEL; and RAPHAEL FINKEL. 2018. Kratylos: A tool for sharing interlinearized and lexical data in diverse formats. *Language Documentation & Conservation* 12.124–146.

LEONARD, WESLEY Y. 2017. Producing language reclamation by decolonising 'language.' *Language Documentation and Description* 14.15–36. doi:10.25894/LDD146.

LUKANIEC, MEGAN. 2022. Managing Data from Archival Documentation for Language Reclamation. *The Open Handbook of Linguistic Data Management*, ed. by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 315–326. The MIT Press. doi:10.7551/mitpress/12200.003.0030.

DE MARNEFFE, MARIE CATHERINE; CHRISTOPHER MANNING; JOAKIM NIVRE; and DANIEL ZEMAN. 2021. Universal Dependencies. *Computational Linguistics* 47.255–308. doi:10.1162/coli_a_00402

MACWHINNEY, BRIAN. 2000. *The CHILDES project: tools for analyzing talk*. 3rd ed. Mahwah, NJ: Lawrence Erlbaum.

MATTERN, ELEANOR. 2022. The Linguistic Data Life Cycle, Sustainability of Data, and Principles of Solid Data Management. *The Open Handbook of Linguistic Data Management*, ed. by Andrea L. Berez-Kroeker, Bradley McDonnell, Eve Koller, and Lauren B. Collister, 61–72. The MIT Press. doi:10.7551/mitpress/12200.003.0009.

MAXWELL, MICHAEL. 2003. *Hermit Crab: parsing and generating with classical generative phonology and morphology*. Dallas: SIL International Digital Resources. https://www.sil.org/resources/archives/5746.

MCCARTY, TERESA L.; SHEILAH E. NICHOLAS; KARI A. B. CHEW; NATALIE G. DIAZ; WESLEY Y. LEONARD; and LOUELLYN WHITE. 2018. Hear Our Languages, Hear Our Voices: Storywork as Theory and Praxis in Indigenous-Language Reclamation. *Daedalus* 147.160–172. doi:10.1162/DAED_a_00499.

MEAKINS, FELICITY; JENNIFER GREEN; and MYFANY TURPIN. 2018. *Understanding linguistic fieldwork*. London: Routledge.

MIROMAA ABORIGINAL LANGUAGE & TECHNOLOGY CENTRE 2023. Miromaa. https://www.miromaa.org.au/miromaa-software/get-started.

MOSEL, ULRIKE. 2015. Putting oral narratives into writing – experiences from a language documentation project in Bouganville, Papua New Guinea. *Language Contact and Documentation / Contacto lingüístico y documentación*, ed. by Bernard Comrie and Lucía Golluscio. Berlin: De Gruyter. doi:10.1515/9783110317473.321.

MUSGRAVE, SIMON; and NICK THIEBERGER. 2021. The language documentation quartet. *Proceedings of the 4th Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, 6–12. Online: Association for Computational Linguistics. https://aclanthology.org/2021.computel-1.2.

RAY, ERIK T. 2003. *Learning XML*. Second edition. Beijing, Köln: O'Reilly.

REHM, GEORG. 2016. The Language Resource Life Cycle: Towards a Generic Model for Creating, Maintaining, Using and Distributing Language Resources. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2450–2454. Portorož, Slovenia: European Language Resources Association (ELRA). https://aclanthology.org/L16-1388.

ROGERS, CHRIS. 2010. Review of FieldWorks Language Explorer (FLEx) 3.0. *Language Documentation & Conservation* 4.78–84.

SANDE, HANNAH; and MARCUS EWERT. 2023. TwistedTongues. https://twisted-tongues-beta.appspot.com/#/.

SKILTON, AMALIA; SUNKULP ANANTHANARAYAN; SOFIA PIERSON; and CLAIRE BOWERN. in press. Flibl: A tool to ease text transfer between ELAN and FLEx. *Language Documentation & Conservation*.

SNELLER, BETSY; SUZANNE EVANS WAGNER; and YONGQING YE. 2022. MI Diaries: ethical and practical challenges. *Linguistics Vanguard* 8.307–319. doi:10.1515/lingvan-2021-0051.

SNELLER, BETSY; RUSS WERNER; HEMKESH AGRAWAL; DANIEL MAGAWAY; TESS COLEMAN; SHIYU YAN; and ALBERT CROOKS. 2023. MiDRF App Builder. https://builder.mi-diaries.org/.

SULLIVANT, RYAN. 2020. Archival description for language documentation collections. *Language Documentation & Conservation* 14.520–578.

SUMMER INSTITUTE OF LINGUISTICS 2022. FieldWorks Language Explorer. https://software.sil.org/fieldworks/download/fw-91/fw-9118/.

SUSLAK, DANIEL. 2005. The future of Totontepecano Mixe: Youth and language in the Mixe highlands. Chicago: University of Chicago dissertation.

TSIKEWA, ADRIENNE. 2021. Reimagining the current praxis of field linguistics training: Decolonial considerations. *Language* 97.e293–e319. doi:10.1353/lan.2021.0072.

WARNER, NATASHA; QUIRINA LUNA; and LYNNIKA BUTLER. 2007. Ethics and revitalization of dormant languages. *Language Documentation & Conservation* 1.58–76. doi:http://hdl.handle.net/10125/1727.

WHALEN, D. H.; MELISSA E. LEWIS; STEFANIE GILLSON; BRITTANY MCBEATH; BRI ALEXANDER; and KATE NYHAN. 2022. Health effects of Indigenous language use and revitalization: a realist review. *International Journal for Equity in Health* 21.169. doi:10.1186/s12939-022-01782-6

WILKINS, DAVID. 1992. Linguistic research under Aboriginal control: A personal account of fieldwork in Central Australia. *Australian Journal of Linguistics* 12.171–200. doi:10.1080/07268609208599475.

WILLIAMS, NICHOLAS; W. D. L. SILVA; LAURA MCPHERSON; and JEFF GOOD. 2021. COVID-19 and documentary linguistics: Some ways forward. *Language Documentation and Description* 20.359–377. doi:10.25894/ldd57

WITTENBURG, PETER; HENNIE BRUGMAN; ALBERT RUSSEL; ALEX KLASSMANN; and HAN SLOETJES. 2006. ELAN: a professional framework for multimodality research. *5th International Conference on Language Resources and Evaluation (LREC 2006)*, 1556–1559.

WOODS, LESLEY. 2023. *Something's Gotta Change: Redefining Collaborative Linguistic Research*. Canberra: ANU Press. doi:10.22459/SGC.2022.