

# LINGSYNC: A FREE TOOL FOR CREATING AND MAINTAINING A SHARED DATABASE FOR COMMUNITIES, LINGUISTS AND LANGUAGE LEARNERS\*

MARYELLEN CATHCART<sup>a,b</sup> GINA COOK<sup>b</sup> THERESA DEERING<sup>b,c</sup> YULIYA MANYAKINA<sup>d</sup>  
GRETCHEN MCCULLOCH<sup>d</sup> HISAKO NOGUCHI<sup>e</sup>

<sup>a</sup>University of Delaware <sup>b</sup>Language Lab <sup>c</sup>Visit Scotland <sup>d</sup>McGill University <sup>e</sup>Concordia University

## Abstract

LingSync is a free, open source data management system built for field linguistics teams. It allows teams to securely enter, store, organize, annotate, and share linguistic data. The application is accessible on any device: not only on laptops (Mac, Linux, Windows, ChromeBooks) but also on touch tablets or mobile devices (Android and iPhone/iPad). It is suitable for both online and offline use, and data is syncable and sharable with other researchers as well as the language community. Team members can use the application not just to view and modify data, but also to analyze and discuss it. The system also has a simple and friendly user interface, allowing users to record audio/video directly into the database. The application has import and export capabilities for multiple file types. LingSync was designed from the ground up to conform to E-MELD and DataOne data management best practices, an important requirement for any application used by data collection projects funded by granting agencies. Finally, the application is designed to be intuitive and theory-free, so it is not necessary to be a field linguist or programmer to figure out how it works. LingSync is hosted on cloud servers so that users can use it without knowing how to set up their own servers, but it also has an installation guide for server administrators so organizations can run their own instance of LingSync. Not only is its source code 100% open, but LingSync's development is also 100% open and driven by the research assistants of fieldwork teams who use the application.

---

\*Presented at CAML as *iCampo: Un aplicación de colección de datos lingüísticos*

We would like to thank the CAML participants for their feedback and questions, as well as sharing their knowledge and fieldwork experiences; without them this project would never have begun. We would also like to thank Android Montréal and JavaScript Montréal for their comments and feedback on technical aspects of LingSync presented at meet-ups. We would like to express our deep thanks to Tobin Skinner, Josh Horner, Elise McClay, Louisa Bielig, Joel Dunham, Brian Doherty, Gay Hazan, Oriana Kilbourn, Rakshit Majithiya, Kim Dan Nguyen, Pablo Duboue, Xianli Sun as well as countless other open source software developers who directly or indirectly helped build LingSync to what it is today and will be in the future. We would like to thank LingSync users for providing feedback, suggestions, asking questions and sending bug reports, all of which have been instrumental driving the software's development. Finally, we would like to thank Jessica Coon, Alan Bale and Michael Wagner for their guidance and challenging us to keep the app simple and yet flexible, as well as SSHRC Connection Grant (#611-2012-0001) and SSHRC Standard Research Grant (#410-2011-2401) which partially funded the students who have doubled as fieldwork research assistants and interns on the project. All errors and oversights are naturally our own.

# 1 Introduction

LingSync was conceived out of the needs of language researchers doing fieldwork or other large-scale data collection, often in a partnership with language community members. Unlike professional field linguists, language researchers often do not work with a database on a full-time basis, but rather as one of many other tasks such as teaching, grading, researching and preparing publications. Linguistic fieldwork frequently occurs where a stable connection to the internet is not guaranteed. In addition, it frequently involves a group of researchers, research assistants and native speakers contributing to a single data collection. An ideal linguistic database should therefore work both online and offline (Wittenburg et al., 2006), as well as make it easy to share and integrate data resources, not only among researchers but also with the language community (Good, 2012, Thieberger, 2012).

Thieberger (2012) identifies “a need for research into existing and emerging methods and development of tools both for creating linguistic data and then for making it useful.” In this paper we hope to demonstrate how the LingSync project embodies Thieberger’s call to action. Section 2 provides some background to the project: we explain in §2.1 why LingSync was created despite numerous existing resources, in §2.2 the recent technology that now makes it possible to create such an application, and in §2.3 the data management best practices that form an essential part of LingSync’s design. Section 3 discusses LingSync’s core §3.1 functionality as well as how LingSync empowers teams with a plugin architecture §3.2 enabling the repurposing of existing resources and software libraries. We conclude in Section 4 with a brief discussion in §4.1 of how teams are currently using LingSync, and celebrate LingSync’s growing community of over 300 users in just 1.5 years since the project was launched at CAML in August 2012.

This paper does not constitute a user guide for LingSync, but rather a discussion of how the LingSync system addresses some of the common challenges facing field linguistics teams. Readers who are interested in *using* the system, not in the underlying system itself, may prefer to jump to section §3.2.1 to see the new interface built based on CAML feedback which is almost as easy to use as Excel or Google Spreadsheets. For up-to-date tutorials and user guides, see the project webpage (LingSync, 2013).

## 2 Background

### 2.1 Existing software: Pros and cons

There are several existing programs designed specifically for storing linguistic data; however, none of them fully satisfies a fieldwork team’s need for robust, collaborative, multi-platform data annotation and organization, both online and offline. For example, there are web-based databases which allow collaboration and sharing research with the language community, such as the Yurok Documentation Project (Garrett et al., 2001), Karuk Dictionary and Texts (Garrett et al., 2009), the Washo Project (Yu et al., 2005, 2008, Cihlar, 2008), and the Online Linguistic Database (OLD) (Dunham, 2010) to mention only a few. However, they only work online, making them unusable for editing or searching the data while in the field with limited or no internet access.

There are also non-web-based software programs such as ToolBox (SIL International, 2003) and FLEx/FieldWorks (SIL International, 2011), which are often the best tools for annotating data and organizing data into various forms of deliverables including a corpus, grammar and lexicon.

However, these offline tools were not designed for collaboration. Field workers generally each enter data on a single computer, and merge data later when online. This means that collaboration using these tools must use an ad-hoc mechanism such as a shared network drive, Dropbox, Google Drive, or email (or in an industry setting, a version control system such as SVN or Git), as well as transformation scripts to permit multiple users to combine their data structures with other fieldworkers who work on related projects. One of the most difficult problems to overcome is that these tools only run on platforms which are popular among professional fieldworkers (usually Windows and sometimes Linux), but not Mac or mobile devices. They can also require too much time and training to be practical for field linguistics labs with a high turnover rate or field methods courses: in both cases, team members do not have time for data management as a full-time job (Butler and van Volkinburg, 2007). Finally, there are many other ad-hoc solutions which are not specifically designed for linguistic fieldwork, including general purpose database software such as FileMaker Pro, which can be customized for the purpose of language research. However, they incur the same problems as other offline tools, and additionally require hiring a programmer or programming-linguist to customize the software for the purpose of linguistic research.

Furthermore, none of the linguistic database software surveyed above provides a modern user experience. The number of clicks required, the delay between actions, and inability to efficiently browse the data did not meet current Human Computer Interaction and Software Engineering best practices, not to mention the expectations of users who are accustomed to using professionally crafted, data-heavy software such as Facebook, WordPress, Evernote and Google. Field linguistics teams have very limited resources. A good user experience is absolutely crucial to maximize the amount of high-quality data produced for the budgeted research hours (Palmer, 2009). LingSync grew out of discussion with a number of fieldwork labs who had improvised their own ad-hoc systems, despite being aware of the existing packaged professional fieldwork solutions mentioned above, and programming-linguists who worked in the software industry, and knew that the fabric of technology had changed significantly enough to make it feasible to build a system which could reconcile many of the previously irreconcilable constraints.

## 2.2 Technological background: Why we can make this now

While the rest of this paper (with the exception of Plugins §3.2) targets a field linguist audience, a discussion of LingSync is incomplete without some discussion of why LingSync is unique among field databases. LingSync is actually only an open ended data schema (as shown in (4) below) and a few simple Node.js web services. All of the heavy lifting of the system is done by CouchDB. CouchDB is an open source project which began in 2005 and matured in 2010, with continuing exciting additions each year following.

CouchDB solves several important problems for field linguists, which are summarized in (1). The first of these problems, as shown in (1a), is the ability to have highly complex data structures yet easily change these data structures as the field database evolves and matures.<sup>1</sup> Secondly (1b), CouchDB was designed for collaboration, as well as to work offline.<sup>2</sup> With CouchDB under

<sup>1</sup>CouchDB stores data as JSON, which is equivalent to XML, meaning that it is more similar to the flexibility of ToolBox and ELAN but yet still has the ability to be extremely structured like FLEx.

<sup>2</sup>All documents are versioned, so each time a team member saves a record, it gets a new version. CouchDB knows how to keep two or more computers in complete sync, permitting team members to go offline with a full copy of their data and come back online again without manually merging all of the data.

the hood, teams can prototype iPhone and Android apps and other useful results *with* language communities (1c) *while* doing fieldwork, without waiting to export their data to another tool.<sup>3</sup> At the same time, all the permissions of the data are still enforced, meaning that the same database can not only serve primary audio and video, but also keep it private until it has been polished for wider community use.<sup>4</sup> CouchDB can look like a webpage, with colors and buttons (1d) but the team's power users can seamlessly explore data in its true form, and even edit and customize existing LingSync scripts to clean and transform data into new shapes for analysis and export.<sup>5</sup>

(1) Previously Irreconcilable Constraints

		Year	Technology
a.	highly structured, yet flexible	2005	NoSQL
b.	collaborative, yet offline	2012	IndexedDB
c.	open data enabled, yet respects the wishes of language community members	2012	CORS
d.	easy to use, yet provides a seamless context switch to a power-user friendly interface	2005	Chrome DevTools CouchDB

## 2.3 Design principles: E-MELD-compliant

The principal goal of LingSync is to help language researchers collect and organize linguistic data and to facilitate collaborative research work. Its main objectives are outlined in (2).

(2) Objectives

- a. A self-explanatory, easy-to-use user interface so that team members can understand and start using the application without laborious training about the software.
- b. Customizable data entry fields to accommodate particular requirements of a research.
- c. Data sharing, protection and integration functions to facilitate collaboration among researchers and between researchers and language consultants.

LingSync is designed so that it requires no training or complicated set-up for data categories, and takes no time to add new categories. The application does not include any theoretical constructs that must be tied to the data, but rather allows data fields and categories to develop organically as

<sup>3</sup>CouchDB is, in fact, not just a database, but also a server. It is able to serve data at a unique URL without the need to write a custom API.

<sup>4</sup>CouchDB uses authentication tokens which can be shared across different client apps, meaning that if the user logs in to LingSync in one window, they can access the data in another app in another window (an experience very similar to Gmail and Google Docs). LingSync adds additional measures discussed in Sharing §3.1.4 below.

<sup>5</sup>Most of the labs we talked with had someone in their lab who had learned Python to be able to transform data. Both Chrome and CouchDB have an interactive scripting console where you can script transformations to your data and visually preview the results immediately, in color and with a mouse. Since the user interface is in the browser and uses only JavaScript, it is much easier for research assistants to learn to use than a text only terminal interface and traditional programming languages. Research assistants can copy-paste their way to new scripts and other useful internal tools for their team, with no need to train an external technical consultant in the subtleties of linguistic data to ensure that the way they clean data doesn't introduce errors (or overregularization) which was not present in the underlying data.

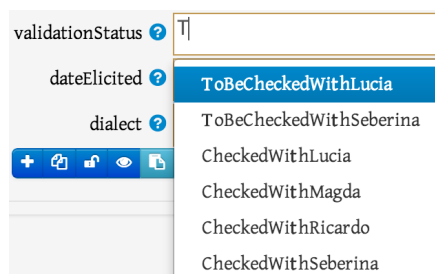
data collection proceeds. Researchers are able to add and change their fields and categories for the data at any point.

In addition to the user-driven objectives in (2), LingSync was designed to comply with the recommendations from E-MELD Best Practices in Digital Language Documentation (E-MELD, 2006b) and DataONE Primer on Data Management (DataOne, 2011).<sup>6</sup> E-MELD describes seven common problems in digital language documentation identified in Bird and Simons (2003). The sections below discuss how LingSync implements solutions for each identified problem.

### 2.3.1 **Content:** Data is annotated and described using consistent terminology

LingSync allows users to design their annotation conventions per corpus and makes these conventions available as help text next to each field of each record of the corpus. The easily-available help text ensures that conventions are never far away, and reduces tension among team members. Although a consistent terminology is necessary for cross-corpora integration (Schalley, 2012), since each team of users works within a slightly different linguistic framework or set of conventions, no pre-determined terminology is sufficiently flexible to meet the needs of all users. LingSync therefore permits researchers to use open-ended categories for annotation, but to enable consistency, users see typeahead drop-down menus containing terms previously used by their team members, as shown in (3). At this point, users may accept existing terms from the drop-down, or add new categories without needing to leave the data entry interface.

(3) Data is open-ended, yet consistent



The screenshot shows a data entry form with several fields. The 'validationStatus' field has a question mark icon. The 'dateElicited' field has a question mark icon. The 'dialect' field has a question mark icon and a dropdown menu. The dropdown menu is open, showing a list of terms: 'ToBeCheckedWithLucia', 'ToBeCheckedWithSeberina', 'CheckedWithLucia', 'CheckedWithMagda', 'CheckedWithRicardo', and 'CheckedWithSeberina'. Below the dropdown menu is a row of icons: a plus sign, a document icon, a magnifying glass, and a trash can icon.

### 2.3.2 **Format:** Data is intelligible regardless of the type of operating system

LingSync is built using web technologies which permit the application to be 100% Unicode, including support for right-to-left orthographies as well as IPA symbols. LingSync data is stored as plain text, and exportable in human-readable formats including JSON, LaTeX, CSV, and XML, all of which are non-proprietary and compatible with applications commonly used in linguistic data collection and documentation. LingSync runs on Mac OS X, Linux and Windows, as well as on newer platforms such as Android, ChromeBook and iOS (iPhone/iPad).

(4) Data is stored as plain text in Unicode<sup>7</sup>

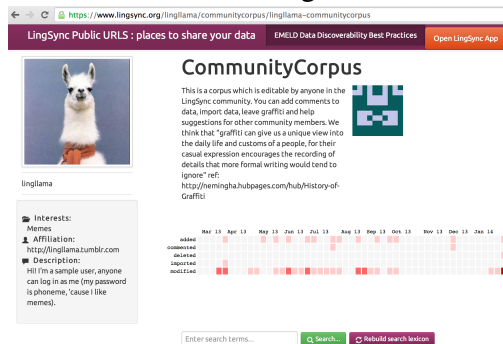
<sup>6</sup>For brevity only EMELD practices are addressed here, for details on how LingSync addresses DataOne recommendations see the WhitePaper (LingSync, 2012c).

<sup>7</sup>For more examples of LingSync's data schema see <https://github.com/OpenSourceFieldlinguistics/FieldDB/tree/master/tests>



(6)

a. Data can be search engine discoverable



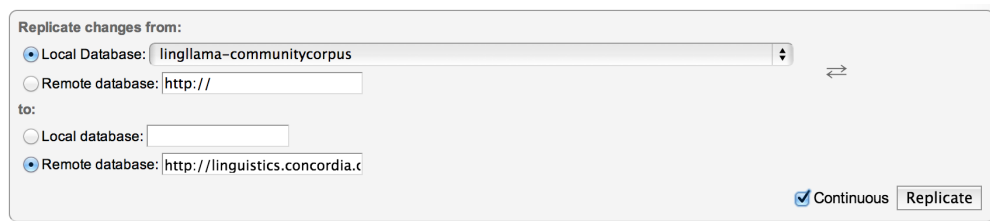
b. Fields can be public yet masked



### 2.3.4 Access: Data is accessible

Much of linguistic data sits in researchers' offices in the form of notebooks or tapes, or as files on their computers. LingSync data is online, and team members can even replicate the entire database (7) onto their laptops or department servers. Unlike tape data and note cards, the data can be in many physical locations at one time, and even available to collaborators across the world.

(7) Data can be fully replicated in its entirety across multiple servers



### 2.3.5 Citation: Database provides citation information

In LingSync, each record and media resource of each corpus has a unique URL, shown in (6a) above, which can be used for citation. All data points are tied to an Elicitation Session, which contains details of data source. The source is often language consultant(s), as shown in (6b) but may also be a publication, historical text, media broadcast or web page. Other important information include the date when the data was elicited/uttered, as well as other metadata as determined by the team conventions to ensure that data quality can be traced to its primary source.

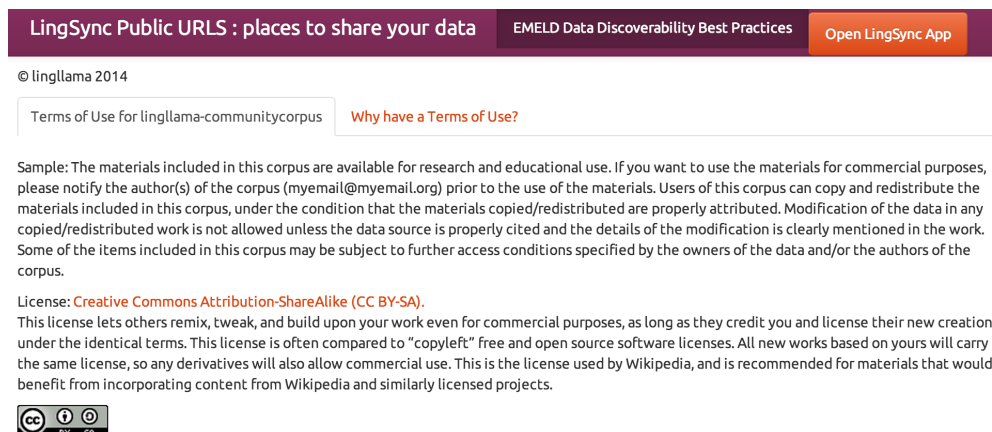
### 2.3.6 Preservation: Data are archived in a way that withstands long-term preservation

Preservation of digital data is always confronted with the possibility that the data file format over time becomes obsolete and unsupported by new technology. LingSync stores data in a plain text JSON format as shown in (4) above. JSON files are equivalent to XML: they are lightweight text files in which data contents are easily readable by humans and scripts. In addition to a host server, the data can be stored in multiple locations, as we can see in (7) above. Corpus administrators can schedule regular archiving by creating a "bot" which will archive their data to one of the existing, reputable language archives of the user's choosing (E-MELD, 2006a).

### 2.3.7 **Rights:** Rights of authors of data and of language consultants are respected

Each corpus includes a section for the Terms of Use (8) where authors of data can specify conditions for data usage. Corpora often contain sensitive information, consultant stories and other information which must be kept confidential. In LingSync, data and information designated as confidential are encrypted prior to storage in the database using the United States Federal approved AES encryption standard, and are presented as masked data, as shown in the ‘gloss’ field of (6b) above.

#### (8) Teams can have custom terms of use



## 3 Features of LingSync

LingSync has two main types of functionality: functionality for linguistic field databases, and functionality for user-friendly, community-driven software. In this section, we describe the features by which LingSync meets field linguists’ goals, glossing over some of the software engineering practices which make LingSync a usable and maintainable system.<sup>8</sup>

Readers who are interested in detailed instructions on how to use LingSync may prefer to consult the project’s website (LingSync, 2013) for the latest information and tutorials, or search for ‘LingSync’ on YouTube<sup>9</sup> to watch video demos and tutorials.

### 3.1 Core Functionality

#### 3.1.1 Data entry and import

Data entry in LingSync goes beyond just typing or transcribing data. While simply typing in or importing data is the most common use case, LingSync also provides the ability to add comments to any data in the system. This makes it possible to collaboratively enter and discuss data, without modifying or destroying information in the data itself. This means that multiple team members can suggest new segmentation, new gloss information, or qualms about translation or context, and the team can reach a consensus together without blocking team members from accessing or improving data. Comments are also searchable, editable and deletable, and can be formatted using

<sup>8</sup>For a complete list of functionality see the Software Design Description (LingSync, 2012b).

<sup>9</sup>[http://www.youtube.com/results?search\\_query=lingsync](http://www.youtube.com/results?search_query=lingsync)



wiki markup, which permits users to add unlimited documentation to a data record without needing to put the documentation into the record's translation or a dedicated notes field. When working as teams composed of linguists or community members who may speak different dialects and thus have differing judgements, we believe that comments are a key way in which teams can provide a maximum amount of access and curation, without worrying about different team members over-writing each other's judgements.

Since every previous version of every document in the system is saved, mis-guided edits by team members can be detected via the team activity feeds and undone if desired (discussed in §3.1.4). If a team discovers one of its members is not following their team's data curation conventions, the permissions system, discussed in §3.1.4, allows the team to set that individual's permissions to read and comment only.

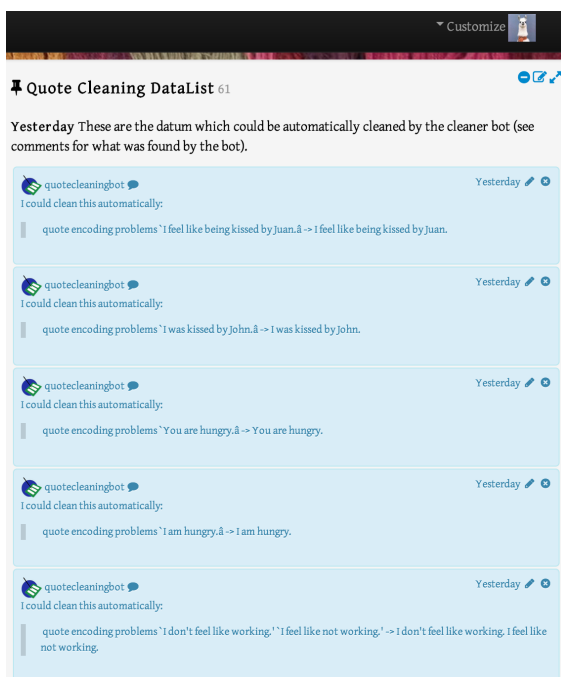
Data can be categorized using tags as well as by validation status of each individual data entry. A record could have a simple status such as "Checked," or a more complex status such as "CheckedWithSeberina" or "ToBeCheckedWithConsultant" if the team is working with multiple consultants and/or dialects (shown in (3) above). The ability to group data into its validation status further aids organization and permits the team to plan data for future elicitation sessions, or to send data to consultants to be checked either by exporting the data and sending it by email, or by adding the consultant as a team member.

LingSync is "skinnable," which means that each user can have a different visual representation of the data. Teams can even create non-technical views of the data so that the language consultants can enjoy being part of the team, and feel more connected and included in the collaborative nature of the language documentation effort, without having to see technical fields like morphemes or semantic representations if they do not desire.

In most teams with long standing databases, it is often the custom to enforce conventions by providing users with drop-downs where they must select only from appropriate options, or go to another screen to configure the new option before selecting it. However, Palmer (2009), Cihlar (2008) and Wittenburg et al. (2006) among others state the need for tools to adapt as a fieldwork project matures and analyses change, "Carletta et. al. (2000) argues that linguistic data sets are varied and idiosyncratic to the point where imposing a universal annotation/description scheme would be impractical and counterproductive" (Cihlar, 2008:p.11). Wittenburg argues that a software's "ergonomic qualities greatly contribute to the experience and appreciation of the every day user. Also the level of productivity that can be reached is of utmost importance." (Wittenburg et al., 2006:p.1559). Rather than enforcing universal data conventions, or making heavy use of drop-downs, LingSync makes use of typeahead and autocomplete, and makes data entry and data curation also fully scriptable. Since a fundamental part of fieldwork is exploring and re-analyzing data, LingSync allows users to create bots which partly automate cleaning tasks. Bots can even be scheduled to run periodically on the corpus, as shown in (9a), reducing the manual data cleaning process if, for example, the team decides partway through creating a database that all data should use the convention *PAST should be glossed as PERF in the context of ASP*. Bots are able to go through a corpus, and leave comments on data which should be cleaned manually, or even execute the changes (9b) after the team has reviewed and approved the changes.

(9)

a. Bots can recommend changes and



b. leave explanations of what was changed



Data entry is expected to be grouped by elicitation session (or by publication or other data sources). In fact, one expected method of data entry is not data entry at all, but rather the video recording of an elicitation session followed by transcribing the session at a later date.<sup>10</sup> Longer audio/video files can also optionally be uploaded to the speech web service (see §3.2.4) which uses Praat to automatically split the audio stream into utterances. This approach to data entry permits the team to dedicate 100% of their attention to the speaker and formulating questions while eliciting data, rather than dividing their attention between the speaker and the process of data entry. One of LingSync's founding principles is that you should only need to enter data once. Whether you enter it in an Excel Spreadsheet, in a handout, in ELAN or in FLE<sub>x</sub>, you should be able to import it into LingSync without needing to re-enter the data. Each record in a LingSync database can have an unlimited number of fields, with unlimited complexity, making it possible to import other formats, and be able to re-export them without losing any information (for example, timed alignments in ELAN or Praat).

### 3.1.2 Auto-glosser

Similar to the glosser underlying FLE<sub>x</sub> (Black and Simons, 2006), the semi-automatic glosser requires no configuration or set up to be useful. It “learns” from the data in a corpus to guess where words might be segmented (10a), or how morphemes should be glossed. The glosser is also a separate module, which means that if you have an existing glosser, you can plug it in to LingSync. Glossers can also be shared. For example, if you have two Quechua corpora, you can

<sup>10</sup>For example, Cox (Cox, 2013) considers recording speech and its free translation orally provided by language consultants, when balanced with a careful review, as a rapid yet valid annotation method alongside written transcription and annotation.

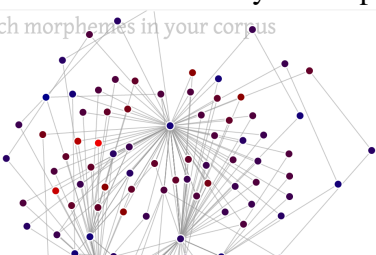
set the glosser URL to use either corpus, or permit other teams to use a glosser which was trained on your corpus.

(10)

a. Morpheme segmentation

utterance ?	noqata tusunaywanmi
morphemes ?	n
gloss ?	noqata tusu-nay-wan-mi
translation ?	noqata tusunaywanmi

b. uses precedence relations in your corpus



The glosser is designed to make the app “smarter” and to reduce the amount of time spent entering predictable information such as glosses. The glosser can use any existing morphological analysis tool (§3.2) to break down the utterance/orthography line into a probable morphological segmentation using known morphemes in the lexicon, and enter a probable gloss for the morphemes in the glossing line. The glosser module is designed to reduce redundant data entry, not to provide complete glosses. It is, of course, crucial that predicted morpheme segmentation and glosses be corrected by users, particularly in languages that have many short or ambiguous morphemes, as these will result in more possibility for error in automatic morpheme segmentation.

The glosser uses the auto-generated lexicon to evaluate morphemes both by precedence relation and by gloss. Each corpus has its own lexicon, which is loosely modelled after a mental lexicon,<sup>11</sup> as a network of morphemes, allomorphs, orthographie(s), glosses and translations.

### 3.1.3 Search

LingSync was designed for powerful search. As shown above, users can search a corpus (5), or across multiple corpora (6). Similar to ELAN (Wittenburg et al., 2006), the results of searches can even be saved as Data Lists which can be sorted and saved for later exporting, curating for a handout or curating language learning lessons for heritage speakers.

Search can be as simple as a keyword search, which will search the entire record, or involve more complex operations such as search within only a single field, or with one keyword in one field and another keyword in another field. For those users who like to think in Set Theory, LingSync also provides the ability to look at the intersection or the union of search results. For phonologists, LingSync allows search using regular expressions to find segments in context. If there is demand, we have the ability to add search for minimal pairs or to search for phonological features in context using a phonology ontology (a general purpose feature geometry/articulatory feature ontology, or a customized ontology created by the users for their language of interest). Phonological search can also be helpful in preparing stimuli and controlling confounds for psycholinguistic experiments.

Fuzzy search is offered as the default for cross corpora searching via the use of Elastic Search, a powerful open source search engine. Fuzzy search makes it possible to find alternative spellings

<sup>11</sup>It is important to note that LingSync uses a lexicon, not a dictionary. The lexicon is a connected graph similar to theoretical models of mental lexicons, and is useful for linguistic analysis but not useful for producing dictionaries for language communities. LingSync is not suitable for curating dictionaries and the tasks involved in professional lexicography. Instead LingSync offers export to WeSay which makes it possible to build a community driven living dictionary (SIL International, 2007).

across corpora, as is commonly the case for languages where there is no standard writing system, or where dialectal differences mean that one morpheme may have several similar forms across corpora.

### 3.1.4 Sharing corpora and the activity feed

Sharing primary resources and the results of fieldwork is notoriously difficult: “most linguists fail to share their data for reasons ranging from the difficulties involved in curating it into a distributable form, to concerns regarding speaker privacy, to a desire to be finished working with it on their own before giving others access” (Bender and Good, 2010). Even “a funding body like the ELDP cannot get all of its grantees to deposit in an archive in a timely fashion (or at all)” (Thieberger, 2012). As part of the process of preparing data for publication involves collaboration of what data needs to be cleaned or excluded, we hope that providing sharing, cleaning and collaboration features directly integrated in the database will make this process easier for teams.

Corpora in LingSync can be shared as a team, with administrators, who cannot see the data but can add new team members (e.g. a project coordinator); writers, who cannot read the data but can enter new data (e.g. language consultants, or psycho-linguistic experiment participants); readers, who can see the data but cannot edit it (e.g. external collaborators); and commenters, who cannot edit the data but can provide feedback and offer additional information or corrections (e.g. consultants and/or collaborators). Of course, most teams will choose to give all roles to all users, but these roles permit a wider and inclusive data collection team than previously available in other data management tools where the permissions are simply full access or no access.

As a team, users might also want to catch up on recent activity in the corpus. If the corpus is small (only 100 records) users could simply read each record to see what is new, but LingSync also provides team activity feeds which are especially useful for larger corpora or where there is more activity. In the activity feed widget shown in (11), users can see who has modified, commented and created data, as well as who has recorded/uploaded audio, or put records in the trash to be deleted later. There are also user activity feeds which are only presented to the user. A user’s personal feed can help them remember what they were working on last time, particularly if it has been months since they last opened the corpus. For example, the user activity feed could help you remember that you hadn’t finished typing up that elicitation session three months ago before you had to go to class. Given that “[w]e all know of projects which have been completed and for which there are now large datasets that are not being properly maintained” (Thieberger, 2012:p.133), we hope that by introducing activity feeds as a feature of a field linguistics database, teams will return to their data and continue to curate it even after they have created their handouts, given talks and moved on to other aspects of their research.

(11) Activity feeds let team members catch up on recent changes in the database



Thieberger (2012) recommends that systems provide unique URLs so that researchers are able to cite primary resources, which further permits “combining data from disparate sources which could be ‘mashups’ or could, for example, involve correlating transcripts and media” into storybooks, dictionaries and mobile apps such as those created by the Myaamia Center (2001), a partnership between the Miami Tribe of Oklahoma and Miami University. LingSync addresses both concerns by a fine-grained level of sharing where the team controls what is shared and how, while each resource keeps a unique URL.<sup>12</sup> Corpora can also be shared via export in EOPAS XML format and import to EOPAS.org which supports both unique URLs and user friendly browsability.

### 3.1.5 Export

As users of many diverse data management software, we felt it was crucial that LingSync be non-proprietary and open. In a language documentation project, linguistic data must remain usable even when “the delivery system (which could be proprietary software or websites that are no longer maintained) becomes unusable” (Thieberger, 2012:p.132).

One important aspect of this is the ability for teams to export their database in any format they choose, in its entirety, or only portions of data which are relevant to a specific export goal. Teams can even save lists of data for dedicated export purposes, such as data for a handout, or data which they are curating to be published as stories for the language community they are working with. LingSync is also able to export word lists, which can be used either as language learning exercises for heritage speakers or as materials for field methods courses.

Thieberger (2012) points out that “offline use is likely to be most relevant to speakers of the languages recorded, given the lack of affordable – or indeed any – internet access. Such offline use of language records includes printed outputs and media on CD, DVD, or in computer-based (e.g. iTunes) formats.” It is possible to export an entire LingSync corpus including media files either as a ZIP archive, as well as multiple formats such as XML, JSON, plain text, TextGrid, LaTeX and CSV.<sup>13</sup> Like the Washo Project, LingSync seeks “a format that is ‘self-documenting’ so that it will be usable years after when the current technologies used to manipulate the data have long become

<sup>12</sup>Musgrave and Thieberger (2012) discuss the diverse ways a language community can benefit from language documentation and how different communities benefit differently. LingSync’s sharing system permits prototyping ‘mashups’ early in a collaboration that can result in data collection which suits not only language documentation efforts but also the needs of unique language communities.

<sup>13</sup>Import and export from ELAN is quite complex (Schroeter and Thieberger, 2006), currently there are no users using ELAN & LingSync which we know of, so lossless import and export to ELAN is not currently available.

obsolete” (Cihlar, 2008:p.4). When data is exported in its raw form, each field includes the help conventions which were used in the app by teams to tell each other what the field is for. When corpora are exported as a ZIP archive, these help conventions are used to automatically generate a README file which details the corpora’s fields as recommended by E-MELD (see §2.3).

Beyond export, LingSync databases are fully replicable between servers, as shown in (7), which means that team members can have entire copies of their database locally on their laptops, while still remaining in full sync with other team members when they go online. It also means that organizations can back-up their data to their own servers without worrying that data may become stale or out of date.

## 3.2 Plugging into LingSync

One of the strengths of LingSync is that it is built using well-understood web technologies which permit the creation and integration of nearly any existing software as web services.<sup>14</sup> Even complex user interfaces can be combined and integrated with LingSync via the NPM and Bower web module management system. In fact, LingSync itself is composed of numerous plugins which can be re-used or embedded in an organization’s website or existing web based tools.

### 3.2.1 LingSync “spreadsheet”

The LingSync “spreadsheet” interface was launched in May 2013, at McGill University’s Computational Field Workshop.<sup>15</sup> It has a simple user interface designed for field methods classes, and is currently in use by three field methods classes, as discussed in §4.1. The “spreadsheet” plugin can be included or embedded in any organization’s website, including a field team’s own project page if the team wishes. The spreadsheet interface is currently what powers [app.lingsync.org](http://app.lingsync.org) and is therefore recommended to new users: detailed user guides and tutorials are available on the project website (LingSync, 2013). In this section we provide only a brief overview as a short case study of what the development of a custom data-entry centered plugin might entail.

The spreadsheet interface offers multiple data entry templates suitable for Interlinear Glossed Text (IGT) data. The Compact template (12) has 4 core fields, which by default consist of Utterance, Morphemes, Gloss and Translation. The Full template (13) includes Grammatical Judgement and Tags fields in addition to the 4 core fields. In both templates, users can customize which fields are displayed and where. Additional templates can be added if a team needs more fields, for example in January 2014 a template was added for one of the field methods groups which modified the layout and default fields.<sup>16</sup>

#### (12) “Spreadsheet” compact view

<sup>14</sup>For teams collaborating with Software Engineering or Computer Science departments, there are limitless plugins which port advances in Computational Linguistics (Chen et al., 2011) or Natural Language Processing (NLP) including negation and modality scope taggers (Rosenberg et al., 2010), discourse extraction out of social media (Dubuc and Bergler, 2010), NLP web services for mobile apps (Sateli et al., 2013), handwriting recognition research for low resource languages (Sadri et al., 2007) or even training novel speech recognition systems with open source toolkits such as Sphinx-4 (Walker et al., 2004). Sample services which can wrap Bash/Python/Perl scripts into RESTful web services are provided on the OpenSourceFieldlinguistics GitHub. <https://github.com/OpenSourceFieldlinguistics>

<sup>15</sup><http://migmaq.org/workshops/2013-computational-fieldworkshop/>

<sup>16</sup>As the templates are simply HTML, often TAs or students are able to create the template and send it to the LingSync team to be reviewed and included in the plugin. Templates usually take less than one day to build.

Noqata qan qaparinaywanki.	Noqa-ta qan qapari-nay-wanki	me-ACC you-NOM yell-DES-2SG.1OM	I feel like yelling at you.	24-1-2014
Suwanayki Josefina	Suwa-nay-ki Josefina	steal.1-2 Josefina-NOM	--	2-12-2013
<div> <input type="text" value="suwanyaysunki"/> <input type="text" value="suwa-nay-ay-sunki"/> <input type="text" value="3-2OM"/> <input type="text" value="decir una duda, tal vez querer robar"/> </div>				
Payta suwanayan monikita	Pay-ta suwa-naya-n moniki-ta	he-ACC steal.naya.3SG little.animal-ACC	He feels like stealing the little animal	2-12-2013

True to the name, data entry in this interface is spreadsheet-like, complete with keyboard shortcuts for tabbing between cells, and pushing enter to switch between rows. The plugin provides a straightforward, simple user interface which lets users enter data without training, and makes the time spent on data entry roughly equivalent to using Excel or Google Spreadsheets; at the plugin's launch, 20 users each entered 4 IGT data records from an unfamiliar language, resulting in a small corpus of 70+ items in under 15 minutes without any specific instruction or training.

### (13) "Spreadsheet" expanded view

Other features of the spreadsheet interface include the auto-glosser, as described in §3.1.2, which can be turned on and off while entering data, without visiting another screen. The restrictive search function permits users to do keyword search and further filter results by additional search terms. Audio can be recorded and/or uploaded directly into the data records. All data are commentable, and the comment and the audio area can be open (13) or hidden (12), even to a point where the user interface looks similar to a conventional spreadsheet such as Google Spreadsheets (but with the added benefit of a glosser). The spreadsheet plugin was written in Angular.js and is built using Grunt.js. It took roughly three months to build, and six months to beta test.<sup>17</sup>

### 3.2.2 Custom glosser

For a team working on Inuktitut (Bielig, 2013) we wrapped Farley (2012)'s morphological analyzer for Inuktitut, which into a web service, permitting a glosser which was prepared to look up variant surface realizations of morphemes in Inuktitut corpora. The Inuktitut glosser took Farley over five years to build and test, and four days to wrap as a web service.<sup>18</sup>

<sup>17</sup>[https://github.com/OpenSourceFieldlinguistics/FieldDB/tree/master/angular\\_client/modules/spreadsheet](https://github.com/OpenSourceFieldlinguistics/FieldDB/tree/master/angular_client/modules/spreadsheet)

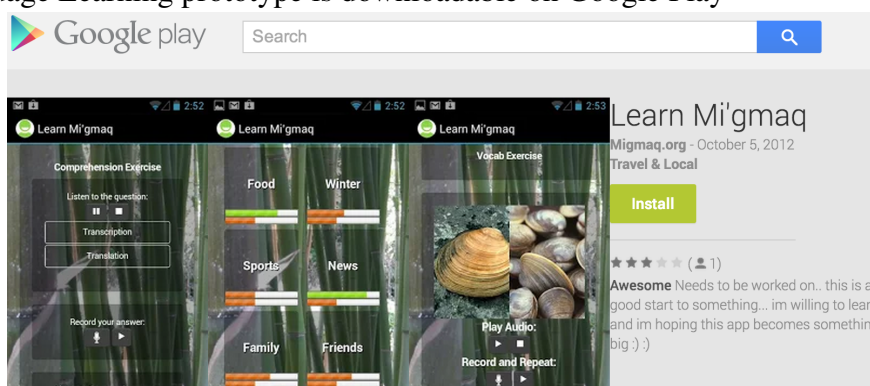
<sup>18</sup><https://github.com/OpenSourceFieldlinguistics/LexiconWebService>



### 3.2.3 Language learning module for Android

We built a prototype mobile app for language learning which uses the LingSync data structure, to enable researchers and language teachers to create language learning aids from the data in existing linguistic corpora and to use data newly collected for the purpose of language learning for linguistic analysis, if desired. The language learning prototype aims to help heritage language learners include their language as part of their daily activities and improve their listening and speaking skills. The orthographic lines (i.e. utterance and morpheme lines) and the attached audio or video recordings of a datum are taken as materials to create lessons. Users are also able to make their own lessons by taking pictures and making audio recordings using their Android phone's built-in features, and share them with other learners. The language learning app took one month to design and implement a clickable prototype.<sup>19</sup>

(14) The Language Learning prototype is downloadable on Google Play<sup>20</sup>



### 3.2.4 Integration with ProsodyLab aligner and touch tablets

The phonetic aligner web service makes it possible to upload audio or video recordings and the orthographic/utterance lines of datum to create a phonetic dictionary unique to the language of the corpus, and to run the McGill ProsodyLab Aligner, a machine learning algorithm which uses Hidden Markov Models to predict boundaries between segments and creates a Praat TextGrid with estimated segments boundaries, saving hours of boundary tagging. The ProsodyLab took roughly two years to build and test the aligner, and it took two weeks to wrap it as a web service.<sup>21</sup>

We also created an Android app which permits recording of high quality video on 7-inch or 10-inch tablets during elicitation sessions. Multiple videos can be associated to a session and uploaded to the ProsodyLab Aligner web service for automatic generation of aligned TextGrids. The android app took roughly one month to build and is currently being beta tested.<sup>22</sup>

<sup>19</sup><https://github.com/OpenSourceFieldlinguistics/AndroidLanguageLearningClientForFieldDB>

<sup>20</sup><https://play.google.com/store/apps/details?id=com.github.opensourcefieldlinguistics.android.lessons>

<sup>21</sup><https://github.com/OpenSourceFieldlinguistics/AudioWebService>

<sup>22</sup><https://github.com/OpenSourceFieldlinguistics/AndroidFieldDBElicitationRecorder>

<https://play.google.com/store/apps/details?id=com.github.opensourcefieldlinguistics.android.videorecorder>



## 4 Conclusion

### 4.1 Users of LingSync so far

LingSync has been used by five field methods classes so far (that we know of), representing a typologically diverse range of languages and a variety of teaching styles. In Winter 2013, it was used at the University of Ottawa (Teenek) and Pomona College (Igikuria). In Winter 2014, LingSync was used at Yale University (Quechua), the University of Connecticut (Nepali) and McGill University (Inuktitut). We especially appreciate the invaluable beta-testing performed by the students and instructors of these classes, results of which are discussed in §3.2.1.

As of February 2014, only 1.5 years after LingSync was launched, there are over 300 users, who have created over 1000 databases. We expect roughly 100 of these users are field methods students, while the remainder are most likely ‘investigating’ accounts despite the availability of the LingLlama user (password: phoneme). We estimate between 10-20 teams (consisting of 1-20 users each) have moved beyond the investigation phase, and have been using LingSync actively. We have been very surprised by the interest in LingSync from field linguists from a variety of locations and linguistic backgrounds, and we gratefully acknowledge the comments and questions from all users, and potential users, which has resulted in substantial improvements in the system.<sup>23</sup>

### 4.2 Summary

In this paper we have discussed some of the challenges inherent in fieldwork projects. We argued that many of the challenges were mutually exclusive and not reconcilable in a single data management system prior to 2012. We have surveyed how some teams have managed to find ways to collaboratively build high quality data archives while balancing their limited resources for research and language documentation, often in collaboration with technical consultants or Computer Science departments. We have presented LingSync, an open source, open-ended system which helps make it easier for fieldwork teams to focus more on data analysis, and less on the repetitive tasks inherent in managing an evolving data heavy project. LingSync was designed from the ground up to be both easy for new users to use, but also respect data management best practices. In the process, LingSync’s open development and use of only one unified, easy-to-learn scripting language for user interfaces, web services and data queries has empowered field linguistics labs to learn more about data automation and build internal technical knowledge, allowing teams to develop novel ways of understanding and exploring their data.

## References

- Bender, Emily M., and Jeff Good. 2010. A grand challenge for linguistics: Scaling up and integrating models. White paper contributed to NSF’s SBE 2020: Future Research in the Social, Behavioral and Economic Sciences initiative.
- Bielig, Louisa. 2013. Supplementing field work with corpora: An investigation of ergativity and anti-passives in Inuktitut. Ms, McGill University.
- Bird, Steven, and Gary Simons. 2003. Seven dimensions of portability for language documentation and description. *Language* 557–582.

<sup>23</sup>Over 1000 features have been requested and implemented since the project began on April 20 2012, for more details, or to vote on features anyone can visit the project’s issue/feature tracker (LingSync, 2012a).

- Black, H. Andrew, and Gary F. Simons. 2006. The SIL FieldWorks Language Explorer approach to morphological parsing. In *Computational Linguistics for Less-studied Languages: Proceedings of Texas Linguistics Society, Austin, TX*.
- Butler, Lynnika, and Heather van Volkinburg. 2007. Review of “FieldWorks Language Explorer (FLEx)”. *Language Documentation & Conservation* 1:100–106.
- Chen, Chenhua, Alexis Palmer, and Caroline Sporleder. 2011. Enhancing Active Learning for semantic role labeling via Compressed Dependency Tree. In *Proceedings of the Fifth International Joint Conference on Natural Language Processing (IJCNLP)*.
- Cihlar, Jonathon E. 2008. Database development for language documentation: A case study in the Washo language. Master’s thesis, University of Chicago.
- Cox, Christopher. 2013. Reclaiming access: Language documentation methodologies in Canadian Indigenous language reclamation. Poster presented at the Canadian Linguistics Association Conference, Victoria, BC.
- DataOne. 2011. Primer on data management: What you always wanted to know but were afraid to ask. URL <http://www.dataone.org/best-practices>.
- Dubuc, Julien, and Sabine Bergler. 2010. Structure-aware topic clustering in social media. In *Proceedings of the 10th ACM symposium on Document Engineering (DocEng ’10)*, ed. Paul Buitelaar, Philipp Cimiano, and Elena Montiel-Ponsoda, 247–250.
- Dunham, Joel. 2010. The OLD. URL <http://www.onlinelinguisticdatabase.org/> SRC: <https://github.com/jrwdunham/old>.
- E-MELD. 2006a. E-meld school of best practice: Finding an archive. URL <http://emeld.org/school/classroom/archives/finding-archives.html>.
- E-MELD. 2006b. E-meld school of best practice: What are best practices? URL <http://emeld.org/school/what.html>.
- Farley, Benoit. 2012. The Uqailaut project. URL <http://www.inuktitutcomputing.ca>.
- Garrett, Andrew, Juliette Blevins, Lisa Conathan, Anna Jurgensen, Herman Leung, Adrienne Mamin, Rachel Maxson, Yoram Meroz, Mary Paster, Alysoun Quinby, William Richard, Ruth Rouvier, Kevin Ryan, and Tess Woo. 2001. The Yurok language project. URL <http://linguistics.berkeley.edu/~yurok/index.php>.
- Garrett, Andrew, Susan Gehr, Line Mikkelsen, Nicholas Baier, Kayla Carpenter, Erin Donnelly, Matthew Faytak, Kelsey Neely, Melanie Redeye, Clare Sandy, Tammy Stark, Shane Bilowitz, Anna Currey, Kouros Falati, Nina Gliozzo, Morgan Jacobs, Erik Maier, Karie Moorman, Olga Pipko, Jeff Spingeld, and Whitney White. 2009. Karuk dictionary and texts. URL <http://linguistics.berkeley.edu/~karuk/links.php>.
- Good, Jeff. 2012. ‘Community’ collaboration in Africa: Experiences from northwest Cameroon. *Language Documentation and Description* 11:28–58.
- LingSync. 2012a. Feature tracker and Milestones. URL <https://github.com/OpenSourceFieldlinguistics/FieldDB/issues>.
- LingSync. 2012b. Software Design Description: Technical specifications & general information for SOEN interns. URL <http://OpenSourceFieldlinguistics.github.io/FieldDB/#sdd>.
- LingSync. 2012c. WhitePaper. URL <http://OpenSourceFieldlinguistics.github.io/FieldDB/#whitepaper>.
- LingSync. 2013. LingSync.org. URL SRC: <https://github.com/OpenSourceFieldlinguistics>.
- Musgrave, Simon, and Nick Thieberger. 2012. Language description and hypertext: Nunggubuyu as a case study. In *Electronic grammaticography*, ed. Sebastian Nordoff, 63–77. Honolulu:

- University of Hawaii Press.
- Myaamia Center. 2001. Myaamia center: Advancing the research needs of the miami tribe of oklahoma with a focus on myaamia language, culture and history. URL <http://myaamiacenter.org/publications/>.
- OLAC. 2006. Open language archives community. URL <http://linguistlist.org/olac/index.html>.
- Palmer, Alexis M. 2009. Semi-automated annotation and Active Learning for language documentation. Doctoral Dissertation, University of Texas at Austin.
- Rosenberg, Sabine, Halil Kilicoglu, and Sabine Bergler. 2010. CLaC labs: Processing modality and negation. working notes for QA4MRE pilot task. In *CLEF Online Working Notes/Labs/Workshop*, 26–30.
- Sadri, Javad, Sara Izadi, Farshid Solimanpour, Ching Y Suen, and Tien D Bui. 2007. State-of-the-art in farsi script recognition. In *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, 1–6. IEEE.
- Sateli, Bahar, Gina Cook, and Rene Witte. 2013. Smarter mobile apps through integrated Natural Language Processing services. In *Proceedings of the 10th international conference on mobile web information systems (MobiWIS)*, ed. F. Daniel, G. A. Papadopoulos, and P. Thiran. Heidelberg: Springer.
- Schalley, Andrea C. 2012. TYTO – a collaborative research tool for linked linguistic data. In *Linked data in linguistics. representing and connecting language data and language metadata*, ed. C. Chiarcos, S. Nordhoff, and S. Hellmann. Heidelberg: Springer.
- Schroeter, R., and N. Thieberger. 2006. EOPAS, the EthnoER online representation of interlinear text. In *Sustainable data from digital fieldwork*, ed. Sebastian Nordoff. Sydney: University of Sydney.
- SIL International. 2003. Toolbox. URL <http://www.sil.org/computing/toolbox/>.
- SIL International. 2007. WeSay. URL [http://www.sil.org/resources/software\\_fonts/wesay](http://www.sil.org/resources/software_fonts/wesay).
- SIL International. 2011. FLEx 7. URL <http://fieldworks.sil.org/download/movies> SRC: <https://github.com/sillsdev>.
- Thieberger, Nick. 2012. Using language documentation data in a broader context. In *Potentials of language documentation: Methods, analyses, and utilization*, ed. Frank Seifart, Geoffrey Haig, Nikolaus P. Himmelmann, Dagmar Jung, Anna Margetts, and Paul Trilsbeek. Honolulu: University of Hawai'i Press.
- Walker, Willie, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. *Technical Report SML1 TR2004-0811*.
- Wittenburg, P., H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. 2006. ELAN: a professional framework for multimodality research. In *Proceedings of LREC 2006, Fifth International Conference on Language Resources and Evaluation*.
- Yu, Alan, Ryan Bochnak, Katie Franich, Özge Sarigul, Peter Snyder, Christina Weaver, Juan Bueno-Holle, Matt Faytak, Eric Morley, and Alice Rhomieux. 2005. The Washo project. URL <http://washo.uchicago.edu/dictionary/dictionary.php> <http://lucian.uchicago.edu/blogs/washo/>.
- Yu, Alan, Ryan Bochnak, Katie Franich, Özge Sarigul, Peter Snyder, Christina Weaver, Juan Bueno-Holle, Matt Faytak, Eric Morley, and Alice Rhomieux. 2008. The Washo mobile lexicon. URL <http://washo.uchicago.edu/mobile/>.