Developing collection management tools to create more robust and reliable linguistic data

Gary Holton
Department of Linguistics
University of Hawai'i
holton@hawaii.edu

Kavon Hooshiar
Department of Linguistics
University of Hawai'i
kavon@hawaii.edu

Nicholas Thieberger
Department of Linguistics
University of Melbourne
thien@unimelb.edu.au

Abstract

Lack of adequate descriptive metadata remains a major barrier to accessing and reusing language documentation. A collection management tool could facilitate management of linguistic data from the point of creation to the archive deposit, greatly reducing the archiving backlog and ensuring more robust and reliable data.

1 Introduction

One of the greatest barriers to accessing language documentation materials is not the lack of standard data formats or archive infrastructure, but rather the lack of descriptive metadata. The 2016 *Language Documentation Tools and Methods Summit* identified a collection management tool as a priority need for documentary linguistics.[1] In response we outline a vision for a collection management tool which will enable linguists to create and manage descriptive metadata from the point of data collection to the point of archive deposit.

The purpose of language documentation is to create and maintain a record of the world's languages and their use (Woodbury 2003). This record is not intended to be locked away on a shelf or a hard drive but rather to be used for further research by future generations of scholars and community members. The record of language documentation should thus be "multipurpose," able to be used for a variety of possibly unanticipated purposes (Himmelmann 2006). Thus, the concept of reuse is a foundational principle of language documentation and arguably one which lies at the heart of linguistics more broadly. To the extent that linguistics is a data driven science, the field relies crucially on access to primary language data.

However, while linguists have always relied on language data, they have not always facilitated access to those data. Linguistic publications typically only include short excerpts from data sets, often without citation (Gawne et al. 2015). There is no single explanation for the slow uptake of archiving and open science among linguists, but three types of barriers stand out, namely:

- lack of archiving infrastructure
- lack of data citation standards and best practices
- lack of appropriate tools

Lack of archiving infrastructure impedes access, since each repository has its own protocols and access restrictions. Lack of citation standards impedes access since researchers have little incentive to share data if they have no guarantee of receiving appropriate attribution. And the lack of tools impedes access by making it difficult to collect, organize, and search language data.

Over the past decade enormous progress has been made to address the first two of these barriers. Yet in spite of these advances in archiving infrastructure and citation practices, the upsurge in data sharing within linguistics has been relatively low. Even among those who are philosophically supportive of open data, there remain significant bottlenecks to actually getting those data into an appropriate archive. We believe the most serious bottleneck concerns the lack of appropriate tools for managing linguistic data. While no two linguistic documentation projects are alike in all aspects, the tools for analyzing field data have become fairly standardized over

[1] https://sites.google.com/site/ldtoolssummit/

the past few decades. The details of the workflows may differ, but the basic approach is common to most documentation projects. However, the management of digital files varies significantly across different projects and across different stages of the same project.

File systems and naming conventions are often developed on an ad-hoc basis and may go through several stages of evolution throughout the course of a documentation project. Metadata may be recorded in a variety of different ways, e.g., in a spreadsheet, a dedicated metadata editor, a text document, a field notebook, or a custom database. Depositing these data into an archive thus requires the linguist to reorganize data, file names, and descriptive metadata in order to satisfy the requirements of the receiving archive. And because different archives require different deposit formats, the linguist must in some cases repeat this process multiple times. For example, a researcher receiving funding from multiple sources may have to satisfy multiple archiving requirements. As a result even well-intentioned researchers may postpone or even forgo archiving altogether. What these researchers lack is a tool to assist with the organization of their collections of data and metadata. While some useful tools have been developed, such as SayMore and CMDI Maker, the lack of uptake among the community of documentary linguists suggests that more development work is needed.

By improving the dialogue between language documenters, language archivists, and developers, this project will serve as a model for the development of linguistic software. The collection management tools in particular will lead to greater uptake of linguistic archives and thus greater availability of language documentation. Most crucially, the collection management tools will lead to better metadata description, as field linguists will be able to enter metadata at the time of file creation rather than after the fact. This improved metadata will in turn lead to greater accessibility and discoverability of language data. This greater availability of primary language resources will transform not only various subfield of linguistics, but also related fields such as anthropology and social psychology, which rely on careful management of field data

2 Version control

Language documentation is an ongoing process, often consuming decades or lifetimes. Traditionally, archiving took place only at the end of a researchers career or following their passing. The obvious advantage to waiting to archive is that one can be certain that all work has been completed. No future versions of materials will be created by the researcher. But the disadvantages are equally obvious and are of two primary sorts.

First, waiting to archive makes the material inaccessible to other researchers for a long period of time. This decreases the efficiency of language documentation since other researchers cannot easily discover what documentation exists for a particular language. Moreover, since linguistic research typically generates vastly more data than can be compiled and analyzed by a single researcher, waiting to archive fails to take advantage of existing expertise. For example, a researcher interested in discourse phenomena may collect vast amounts of recordings which could be relevant to phonetic research but which will not be available to phoneticians until the material is archived. Waiting to archive thus greatly delays the repurposing of linguistic data. This delay is especially salient in cases where the materials may be of use to language maintenance efforts.

A second problem with delaying archiving is that it can be extremely difficult to create descriptive metadata decades after the initial research was done. This problem is particularly difficult when the researcher is deceased and not available to assist in the creation of metadata. In such cases the process of archiving becomes a research activity itself, requiring significant philological work to uncover the intent of the original research effort. Immediate and continuous archiving ensures that descriptive metadata are created in a timely fashion, with minimal additional effort.

Recognizing the problems inherent in delaying archiving, documentary linguists have overwhelmingly endorsed archiving as an essential part of the language documentation process (cf. Gippert et al. 2006). However, there remain significant barriers to archiving language data in practice. Much of the problem stems from the mismatch between current notions of archiving and the established practices of language documentation. Most language archives have been built from the top-down, with pre-defined assumptions about how depositors and other users should interact with the archive. But there is great need to understand the ways in which linguists actually interact with archives. As part of the development process for the Computational

Resource for South Asian Languages (CoRSAL), a new language archive under construction, students at the University of North Texas studied the needs of potential archive users and discovered that depositors may not be well served by traditional archives. Their report states:

> "The concept of an 'archive' and its associated practices are a poor fit with the work practices of linguist depositors. While the logic of archiving requires the deposit of a completed, unchanging artifact, linguists engage in a never-ending process of updating and revising their transcriptions and annotations." (Wasson et al. 2017)

This statement speaks to the need for some kind of version control which allows depositors to archive materials but continue to interact with and engage with those materials as their research continues.

3 Software design issues

3.1 Data model

Although linguistic documentation projects share numerous features, the need to accommodate specific project-based requirements has resulted in a plethora of ad-hoc, proprietary solutions to linguistic data management (cf. Dima et al. 2012). For this reason data models must be extensible in order to accommodate the needs of individual projects. Nonetheless, there are several core aspects which should be a part of any data model, even though they provide challenges. A fundamental requirement is the need to model the interrelationship of recording sessions, media files, and associated secondary data such as transcripts (Hughes et al. 2004). The data model must also robustly handle incomplete information, such as approximation of birth dates. Finally, the data model must employ an ontology to handle the use of non-standard categories and terminology.

3.2 User interface

One of the failures of much linguistic software is to be found in user interface design. It is tempting to think of the user interface as something "extra" which is added onto the core functionality of the software, but if we are to encourage widespread adoption of software it is critical that we design software that people want to use. Currently, most linguistic software is designed to accomplish a specific task. In contrast, most modern software outside the world of linguistics (i.e., "real" software) is designed to attract users. In other words, in the world of real software the focus is on the user rather than the task. Unfortunately, the task-based approach to software is often encouraged by the discipline and its funding regimes. The task is viewed as the intellectual content and hence the object of focus for academic linguists. In contrast, the user interface is seen as an ancillary or decorative -- not part of the core functionality. We argue that good UI design attracts users and is thus critical to the ultimate success of the software. If you want people to do something, you can enable that with your software, but you have to convince them to actually use your software by making it sufficiently user friendly.

Much linguistic software is particularly clunky when compared to modern commercial products. For example, the Arbil metadata editor requires users to enter dates in a very specific YYYY-MM-DD format, though it provides little guidance as to how the date should be entered (Defina 2014). In contrast, most modern software allows dates to be entered in any format which makes sense to the user. The actual date is then inferred. If a user enters "yesterday" in the date field this can readily be interpreted by checking the current date. If a user enters "22 May" in the date field the software assumes that the current year is intended. If a user enters "May 2012" the software infers that the actual day of the month is unknown or irrelevant and thus stores the date as 2012-05.

There are many precedents for good data management software outside the field of linguistics. One familiar example can be found in Apple's iTunes software, which facilitates management of large collections of music files. iTunes facilitates metadata management without requiring that users be aware of collection management best practices. Users make use of iTunes not because they want to manage metadata for their music files but because they want to listen to music. In fact, the user-friendly nature of the iTunes interface has even inspired the repurposing of iTunes as a collection management tool for linguistics and ethnomusicology (Barwick et al. 2005). Another example of good data management software can be found in image organization tools such as Adobe Lightroom. These tools add an additional level of functionality beyond file and metadata management by allowing users to process files directly in associated tools such as image processing software. It is easy to envision this sort of functionality being added to

a linguistic data management tool, facilitating interchange with annotation tools and audio/video editors.

By attracting users, good user interface design can also force and facilitate good practice. An example of this in commercial software can be found in the suite of Google web apps. Google Gmail popularized a number of novel features such as tagging email messages instead of sorting them into folders. But Gmail also subtly forces users to adopt certain practices, such as organizing messages into threads. Moreover, by explicitly avoiding the creation of a stand-alone client, Gmail forced users to access their email in a web-based environment, thus paving the way for adoption of various related web-based applications that are now ubiquitous. As an example of how this force-and-facilitate concept could be applied to linguistic software can be found in automating the creation of certain metadata. For example, the date of a session can be inferred from the timestamp on the associated media files, and graphical cues such as different font colors can be used to prompt that this date needs to be checked by a human. Users can be prompted to enter missing metadata fields, and consistency checks can identify potential errors. Automation can be further facilitated through machine learning algorithms.

3.3 Open source and open development

Ideally, the development of a collection management tool should be accomplished via a collaborative open source effort. Here we use the term open source in the broadest sense which also includes open development. Many linguistic software projects are open source only in the narrower sense. They share their source code, but they do not provide any mechanisms for other users to contribute to the development. That is, they do not facilitate the development of a user community. In more concrete terms such projects may allow users to fork code from a repository and make changes to that code, but they do not permit the code to be pushed back to the repository. As a result the number of contributors to the development of any particular linguistic software tool remains small, and intellectual efforts remain siloed. Given the limited resources available for linguistic software development, the inefficiencies inherent to this approach are a substantial drawback. In contrast, an open development process will take advantage of an untapped pool of coding abilities among practicing linguists and linguistics students.

3.4 Modern software

Modern software should be built using modern best practices. In part this includes the three features discussed above: implementation of a robust and extensible data model; a user-interface which forces and facilitates good practice; and a reliance on open development processes. Modern software should also be cross-platform, not relying on the use of any particular operating system or hardware. Today such software is often built as a web application. Web applications have many advantages that are specifically relevant to language documentation. Not only do they eliminate reliance on a particular platform or device, they also remove the installation process. They can be designed to be used offline, which is essential for much of fieldwork, but they also facilitate sharing information across networks, which fits the goals of archiving and best practice.

4 Building on existing tools

Existing metadata editors provide a good starting point for development of a collection management tool. Early iterations of linguistic metadata editors were closely tied to specific projects and specific metadata standards. Tools such as Arbil (Withers 2012) serve the needs of those required to use IMDI users but do not extend easily to other metadata formats and have a non-intuitive user interface (Defina 2014).

CMDI Maker is a relatively new tool which attempts to overcome these difficulties by making use of HTML browser-based technology and employing an extensible metadata format (Rau 2016).[2] At present metadata can be created in two formats, CMDI and ELAR, reflecting the metadata standards for The Language Archive and the Endangered Languages Archive, respectively. Since the CMDI standard is extensible, additional schema can ostensibly be created. However, the major drawback of CMDI Maker is that it is limited to metadata creation. The workflow assumes that the researcher has already been maintaining metadata in some other format (spreadsheet, field notebook, etc.); the CMDI Maker tool is then used essentially to translate this metadata into the format required for the archive deposit. It is this extra step of metadata translation which becomes a barrier to the archiving process. More significantly, CMDI Maker focuses too narrowly on metadata rather than on the management of a collection of files, in-

[2] http://cmdi-maker.uni-koeln.de

cluding media, analysis, and metadata. Field workers need to begin managing files from the moment a digital recordings is created on their computer; through to the assigning of descriptive metadata; and on to the addition of analyses such as transcription and other annotation. Ideally this entire ecosystem surrounding the management of the collection would be managed by one tool. The drawback of tools such as CMDI Maker is that they focus too narrowly on metadata entry rather than collection management more broadly.

One existing tool which takes a holistic approach to linguistic data management is SayMore (Hatton 2013).[3] SayMore organizes files directly on the users computer, using a human readable and intuitive directory structure (Moeller 2014). Information about participants is stored in directories named with the participants' names. Information about individual recording sessions is stored similarly according to session name. Metadata is stored in simple human-readable XML files consisting of attribute-value pairs, and these XML files are stored within the relevant directories.

While SayMore does not adhere to any particular metadata schema, the ad-hoc format employed could in theory be ported to any of the commonly used formats. Moreover, because SayMore stores metadata within relevant directories, the entire directory structure could in theory be dumped into an archive as a single deposit while retaining all relevant information. In this way SayMore achieves a crucial disaster-recovery function. Namely, should a researcher become incapacitated or pass away prior to completing an archival deposit, the entire project including media files, analysis files and metadata could be recovered and uploaded without difficulty. This crucial feature is lacking in most other approaches to metadata management.

One drawback to SayMore is that it was designed to run on Windows and cannot be easily ported to other platforms. Moreover, as with much linguistic software SayMore attempts to do too much, including both an annotation tool and a limited respeaking facility. This added functionality is not sufficient to replace dedicated tools such as ELAN and Aikuma, respectively, so it tends to bloat the software and detract from its primary management function. In future it may be possible to more fully integrate a collection management tool like SayMore with other tools, following the Lightroom model discussed above. In the meantime, while SayMore can be considered to be the premier extant tool for collection management, it has yet to be adopted by more than a small percentage for field linguists. Instead most field workers continue to use ad-hoc idiosyncratic methods for managing the collections. Indeed, linguists may not even conceive of their materials as "collections," since they appear more as a conglomeration of disconnected computer files.

5 Conclusion

Management of linguistic data remains a major bottleneck in the language documentation process. Providing better tools for collection management will ease the burden on field linguists and increase the rate of uptake of archiving. As noted by Thieberger & Berez, “our foundations need to be built today in a manner that makes our data perpetually extensible” (2012: 91). A collection management tool will help to strengthen those foundations.

In this short paper we have outlined some desiderata for a collection management tool and suggested ways in which such a tool could be built upon existing foundations. Moving forward, it may well be that that a single solution does not fit all users. However, this is difficult to determine without a better understanding of current practices. In the near future we plan to conduct a collection management survey to assess the range of practices currently employed by linguists. We also envision a series of workshops to bring stakeholders into dialogue regarding the development of a collection management tool.

Acknowledgements

Funding for the June 2016 *Language Documentation Tools and Methods Summit* was provided by the ARC Centre for Excellence in the Dynamics of Language. We are grateful to the participants in the summit for helping to establish the trajectory of this research. We are also grateful to participants in an informal planning workshop held following the Workshop on Open Access at the University of Cologne in October 2016. Current work in progress by the authors to map out desiderata for a collection management tool is support by the US National Science Foundation under grant 1648984.

[3] http://saymore.palaso.org

References

Linda Barwick, Allan Marett, Michael Walsh, Nicholas Reid and Lysbeth Ford. 2005. Communities of interest: Issues in establishing a digital resource on Murrinh-patha song at Wadeye (Port Keats), NT *Literary and Linguistic Computing,* 20**,** 383-397.

Rebecca Defina. 2014. Arbil: Free tool for creating, editing, and searching metadata. *Language Documentation & Conservation,* 8**,** 307-314.

Emanuel Dima, Erhard Hinrichs, Christina Hoppermann, Thorsten Trippel and Claus Zinn 2012. A metadata editor to support the description of linguistic resources. *In:* N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk and S. Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12).* Istanbul: European Language Resources Association, pages 1061-1066.

Lauren Gawne, Barbara Kelly, Andrea Berez and Tyler Heston. Putting practice into words: Fieldwork methodology in grammatical descriptions. *International Conference on Language Documentation and Conservation*, Honolulu, February 28. http://hdl.handle.net/10125/25256.

J. Gippert, Nikolas P. Himmelmann and Ulrike Mosel, editors. 2006. *Essentials of Language Documentation,* The Hague: Mouton de Gruyter.

John Hatton. SayMore: Language documentation productivity. *International Conference on Language Documentation and Conservation*, Honolulu, February 28. http://hdl.handle.net/10125/26153.

Nikolas P. Himmelmann 2006. Language documentation: What is it and what is it good for? *In:* J. Gippert, N. P. Himmelmann and U. Mosel, editors, *Trends in Linguistics: Studies and Monographs 178.* The Hague: Mouton de Gruyter, pages 1-30.

Baden Hughes, David Penton, Steven Bird, Catherine Bow, Gillian Wigglesworth, Patrick McConvell and Jane Simpson 2004. Management of Metadata in Linguistic Fieldwork: Experience from the ACLA Project. *On Language Resources and Evaluation.* European Language Resource Association, pages 193-196.

Sarah Ruth Moeller. 2014. SayMore, a tool for Language Documentation Productivity. *Language Documentation and Conservation,* 8**,** 66-74.

Felix Rau. CMDI Maker – the state and prospects of a HTML5 Web app. *Language Documentation Tools and Methods Summit*, Melbourne, June 1-3.

Nicholas Thieberger and Andrea L. Berez 2012. Linguistic data management. *In:* N. Thieberger, editor *The Oxford Handbook of Linguistic Fieldwork.* Oxford: Oxford University Press, pages 90-118.

Christina Wasson, Gary Holton and Heather Roth. 2017. Bringing user-centered design to the field of language archives. *Language Documentation & Conservation,* 11.

Peter Withers 2012. Metadata management with Arbil. *In:* N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk and S. Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12).* Istanbul: European Language Resources Association, pages 79-82.

Anthony C. Woodbury 2003. Defining documentary linguistics. *In:* P. Austin, editor *Language Documentation and Description, Volume 1.* London: Hans Rausing Endangered Language Project, pages 33-51.