`FileLingR`: An R Script validation tool for depositors and users of digital language collections

Irene Yi
Stanford University
ireneyi@stanford.edu

Claire Bowern
Yale University
claire.bowern@yale.edu

Abstract

In this paper, we use findings from Babinski et al. (2022) to develop a new, user-friendly R script, `FileLingR`, as a collection validation script for both depositors and users of digital archival collections (i.e. a check for depositors before depositing collections in an archive and a way to keep track of collections in progress). Its multiple functions include listing each file and folder name as a way to keep track of a collection's full inventory, showing collection statistics by file extension type to flag missing files (e.g. unmatched file numbers in .eaf and corresponding audio/video), and extracting basic information about ELAN tiers.

1 Introduction

Digital language archives house countless collections of language material and documentation. Such material has always been crucial to language communities and academics for research, revitalization, and reclamation; these collections will only become more and more important with time (cf. Henke and Berez-Kroeker 2016). Reusability and long-term preservation of archival language collections have long been concerns in the field (cf. among others Nathan, 2010; Harris et al., 2015; Bird and Simons, 2003), yet collections still vary in their accessibility, usability, and completeness once they are deposited into archives.

Babinski et al. (2022) found that, in a review of randomly selected digital archival collections, there were a wide variety of issues in opening and using collection files for basic phonetic analysis. Such problems were related to file organization, version control, missing files, difficulty in matching files to metadata, missing metadata, inconsistent transcription tiers in .eaf (ELAN XML transcript files; Wittenburg et al. 2006), among others. Many of these problems arose from the manual compilation of documentation collections, and many could be avoided or improved upon with more consistent practices by individual depositors of collections. To this end, Babinski et al. (2022) provided a list of recommendations for depositors. These included clear file naming conventions, not leaving metadata compilation to the end of a project (a comment long made in fieldwork manuals; cf. Meakins et al. 2018; Bowern 2015), being explicit about the types of materials that must be archived, and not treating the archive as the end point in a documentation life cycle. However, ensuring the integrity and consistency of a documentation collection is difficult to achieve. Software such as LaMeta (Hatton et al., 2021) helps with language documentation projects that are usefully organized around the concept of a "session". But not all fieldwork projects are easily organized in that way.

Feedback from the documentation community indicates that a corpus curation tool will help both users accessing language collections, as well as depositors. In an LSA[1] webinar held in March 2022, Bowern et al. (2022) asked the nearly 60 audience members about difficulties they face in accessing archival materials. The percentages of participants who responded that they faced particular issues are presented in Table 1. All of these issues can lead to significant downstream problems in data processing. Problems with file formatting can lead to the inability of people trying to access archives to even open materials, let alone explore and modify them for any reclamation work they may wish to use them for. With missing or incorrect metadata, we run into issues of attribution, including potentially not limiting access rights appropriately. Another result is that language analysis (and subsequent decisions, such as what to include in language pedagogy) can be muddled when contextual information is unavailable or misleading.

In the process of accessing collections, being unable to find media files that correspond to transcripts due to inconsistent naming conventions can

[1] Linguistic Society of America

Problem	%
File formatting issues	52%
Missing or incorrect metadata	76%
Unable to match transcript files to corresponding audio or visual material	33%
Incomplete collections	71%

Table 1: Feedback from the LSA webinar about issues found in accessing archives

be a major roadblock, as it adds much labor to the load of the language worker trying to prepare the materials for their next stage of use, just trying to discern which files go with which. When someone is faced with an incomplete collection, language work is severely hindered, as the amount of information available is significantly reduced. While the nonexistence of files cannot be remedied, it's sometimes the case that a depositor missed something in the process of preparing a collection, and this simple slip-up leads to a reduction in the possibilities of downstream language work.

Given the importance and prevalence of these issues, the discussion leaders asked what kind of tools would be of practical use to the audience. 81% of the audience said that "a way to keep track of what is in a collection" would be beneficial, and 67% also said so for "a tool which identifies the likely problems in a collection".

`FileLingR`, the R script (R Core Team, 2017) described here, is a first attempt to meet these needs. We use the findings from Babinski et al. (2022) and Bowern et al. (2022) to develop `FileLingR`. This is a set of archival collection validation scripts for both depositors and users of digital archives. `FileLingR` collects and reports information about the contents and structure of a given collection. Crucially, this tool can be used as a check to identify unintentional errorsby depositors before collections end up in archives.

2 `FileLingR` architecture

`FileLingR` is an R Script that is accessed as an R Markdown Document (Rmd) through its GitHub repository.[2] Users specify the collection folder they would like to analyze. There are also a number of package dependencies that `FileLingR` requires, specified in the Rmd file. Users run the setup code block and load the relevant packages.[3]

`FileLingR` uses the following package dependencies:

- `devtools`: facilitates installation of R packages from GitHub. (Wickham et al., 2022)
- `github("dalejbarr/elan")`: parses ELAN files. (Barr, 2015)
- `plyr` : used for ELAN tier parsing functions (Wickham, 2011)
- `magrittr` : installation of the % % function, used in ELAN tier parsing functions (Bache and Wickham, 2022)
- `reticulate` : used in ELAN tier parsing functions (Ushey et al., 2023)
- `xfun` : used for manipulations of filenames: file_ext() to obtain a file extension, sans_ext() to remove a file extension, and with_ext() to change a file extension. (Xie, 2018)
- `yaml` : exports the results to .txt textfiles. (Garbett et al., 2023)

`FileLingR` parses a directory structure and recursively reads in the directory and file names as a text string. It then manipulates those strings to extract file extensions (e.g. .xml, .wav, etc.), creates a dataframe (akin to a spreadsheet) from the segmented text strings, and summarizes the counts of different strings.[4]

3 Functions

`FileLingR`'s functions include providing a way to keep track of a collection's full inventory by listing the entire directory structure with files and folders. This is helpful for `Users` when discovering what is available in a given collection and `Depositors` when validating their own collections to identify potential issues. `FileLingR` uses file extensions to provide collection statistics and flags likely missing files (e.g. unmatched file numbers in .eaf and corresponding audio/video). `FileLingR`'s functionality

[2]The repository link can be found at chirila/FileLingR.

[3]As an anonymous reviewer pointed out, some of the package dependencies that `FileLingR` currently uses have not been updated in more than 5 years. We have forked the repository as an interim solution.

[4]While `FileLingR` does not currently implement this function, it could be easily extended to match metadata strings to a file with additional information. For example, speaker initials could be matched to a lookup table that provides more information about the participants in the research.

also extends to extracting basic information about ELAN tiers to find information about participants in a recording event.

A more explicit set of features is listed below:[5]

- File Counts: lists the number of files of each filetype (by extension).
- ELAN Files: provides a list of EAF files with aligned audio/video transcripts.
- Missing Audio: provides a list of the ELAN transcript files (EAFs) which do not have an associated audio or video file (checks for .wav or .mp4 format).
- Settings Files: provides a list of which ELAN files are missing .pfs/.pfsx files (not usually necessary for loading a project).
- Missing Transcripts: provides a list of the audio files that are missing corresponding .eaf files (and which .eaf files are missing .txt exports).
- Export: exports results to a set of plaintext files, which provide the following:
  - 'Sitemap' – a full list of directories and files.
  - Missing Files – files which are missing matches (audio or transcripts).
  - Matched Files – files which have associated metadata.
  - Filetype Summary – the types of file extensions present in the collection, and the count of each filetype.

4 Further discussion and conclusions

This is Version 1.0 of `FileLingR` and several extensions are currently planned. Extended support of file formats is a high priority. While `FileLingR` lists all file formats in the collection, it only works in detail with .wav and .mp4 files, since these are the most common audio and video filetypes that field linguists use. However, `FileLingR` can be easily extended to include a larger array of audio filetypes.

We plan to implement further information about the contents of ELAN files, such as a list of unique characters used in each tier (useful for identifying misplaced information, questionable transcripts, or orthography type). This can be especially helpful in identifying encoding issues –identifying the use of characters in the Unicode Private Use Area, for example, can help depositors be aware that their transcriptions may not be visible to anyone working on an interface without a font that will render those characters appropriately (cf. SILPUA. We envision a use case where transcripts containing placeholders such as 'XXX' or '???' are identified.

We also plan further extensions that mitigate additional issues identified in Yi et al. (2022) and Babinski et al. (2022); for example, collections that are archived as "working" collections but are missing crucial files, where users make use of non-transferable features. Such cases include using file offsets for ELAN transcripts, which work in the ELAN browser but make other uses of the files impossible (or which don't transfer to other instances of the files).

Further expansions include checking for empty directories (plausibly present for work in progress, but can serve as a more explicit signal to flag problems for projects in final stages before archiving) and conflicts between filenames or minimally differing filenames (such as files with underscores vs. hyphens, files that have suffixes of the form `-final`, `-rev`, `-temp`, and the like). Note that `FileLingR` currently will not search a user's entire hard drive for materials. That is, it assumes that all materials for a documentation project are located in a single, specified directory tree (for the same reason, it will not browse across multiple drives).

Through its functionalities, this script complements tools such as LaMeta (Hatton et al., 2021) in identifying missing or inconsistent information in existing collections. As mentioned earlier, LaMeta makes an assumption about the nature of data in the form of hoping users organize their data around a notion of "sessions." While the LaMeta allows for very broad interpretations of that notion, there are cases in which it is not always appropriate as an organizing principle (e.g. [SOMETHING]). `FileLingR` massages the problem of having to fit into that model by simply looking for file cor-

[5] A reviewer suggested that the sitemap would be more usefully displayed as a tree, and that such a tree can be 'easily' compiled using shell scripts. This is a possible extension, though we question whether such a function would make `FileLingR` more user friendly to audiences who are not very familiar with shell scripts and manipulating code. The current list of files can easily be imported into a spreadsheet, for example, whereas an ascii-drawn file tree is not so straightforwardly manipulated. Some users of `FileLingR` are no doubt familiar with shell scripts, while others are not.

respondences, being ambivalent about specifics regarding directory setup. This at least allows users to perform checks before moving onto further stages, regardless of how the next steps will involve restructuring–users know what all is there, at the very least, and they can find out what might be missing. We plan further integration with LaMeta, by providing support for .meta XML files. In the future we will also give users the option to specify a metadata file with filenames. `FileLingR` would then return (i) a list of files found in the directory that are missing in the metadata file, and (ii) a list of items in the metadata file that are not found in the directory.

We welcome further feature requests from members of the user community, to be submitted on the GitHub site for the project.

An anonymous reviewer suggested that `FileLingR` could be redesigned to be an interface for archives to validate content, as well as for collectors to validate and submit metadata to particular archives. As Sullivant (2020) has noted, linguistic metadata comes in at least five different types. There is *descriptive metadata*, which makes it possible for users to find objects in the collection by describing what is in a file (e.g. the speakers or signers who contributed to its recording, the content of the recording, and the like). *Technical metadata* provides information about the files in the collection, such as their length and format. *Structural metadata* contains the information which links items in collections to one another (that is, providing the information which gives relationships between items), while *preservation metadata* is about the item's status in the collection and its long-term storage and durability. Finally, *rights metadata* gives access information, copyright status, and ownership details.

Within descriptive metadata, there can be information about the participants, the language, the recording context, and other material immediately relevant to the recording event. Metadata itself may contain information that is subject to access rights. For example, it may not be appropriate to release personal information about contributors to a collection. Some types of metadata can be automatically compiled, while others cannot. `FileLingR` can provide technical metadata summaries and can facilitate some types of structural and preservation metadata, but it cannot (and probably should not) attempt to house and manipulate descriptive metadata. As noted above, however, a possible future development is for `FileLingR` to link filenames or abbreviations to a table with explanations, thus augmenting the material in the file structure and making information in collections more searchable and filterable.

Currently, `FileLingR` is implemented in RStudio, and so some familiarity with R and RStudio is required in order to run the program. Users need to be able to identify the relevant directory and knit the RStudio document. They also need to be able to understand the R code enough to make modifications to instructions about what output should be generated if they do not wish to compile all the results that `FileLingR` does by default. We recognize that this may be a hurdle to use for some who would like to use this program.

In the long term, we plan to migrate `FileLingR` to a Shiny app that can be run with more graphical user interaction and that requires little to no familiarity with R. However, given that technical expertise with R is increasingly common in linguistics, we also consider `FileLingR` as a potential entry point for fieldworkers who are seeking further familiarity with R and learn from existing scripts.

We hope that `FileLingR` will assist depositors in preparing their collections for deposit. We also hope it will assist users of archival collections in rapidly inventorying collections so they can find the materials they wish to prioritize.

Ethics Statement

We view accurate archiving as an ethical issue. Particularly for endangered languages, it is a matter of ethics that collections be as complete as possible, particularly where communities and individuals rely on such collections for future language support, revitalization, and reclamation. It is also a matter of good scholarship (Berez-Kroeker et al., 2017, 2018; Harris et al., 2015). As Rice (2022) and Perley (2012) among others have pointed out, the history of academic language documentation has involved extractive practices where linguistic recordings are made, removed from communities, and decontextualized. Metadata and collection structure is crucial to restoring and retaining that context. While language recordings are made and compiled for many purposes, none of these purposes can be met appropriately without sustainable metadocumentation of collections on the behalf of people working with

language materials.

Acknowledgements

Thank you to the Yale fieldwork group (particularly Sunkulp Ananthanarayan and Coralie Cram, and anonymous reviewers for ACL who provided helpful comments.

References

Sarah Babinski, Jeremiah Jewell, Kassandra Haakman, Juhyae Kim, Amelia Lake, Irene Yi, and Claire Bowern. 2022. How usable are digital collections for endangered languages? a review. *Proceedings of the Linguistic Society of America*, 7(11):5219.

Stefan Milton Bache and Hadley Wickham. 2022. *magrittr: A Forward-Pipe Operator for R*. Https://magrittr.tidyverse.org, https://github.com/tidyverse/magrittr.

Dale Barr. 2015. elan. `https://github.com/dalejbarr/elan`.

Andrea Berez-Kroeker, Gary Holton, Susan Kung, and Peter Pulsifer. 2017. Developing standards for data citation and attribution for reproducible research in linguistics: Project summary and next steps. *Presentations from the Linguistic Society of America symposium and poster session on Data Citation and Attribution in Linguistics, 5-9 January 2017, Austin TX*.

Andrea L. Berez-Kroeker, Lauren Gawne, Susan Smythe Kung, Barbara F. Kelly, Tyler Heston, Gary Holton, Peter Pulsifer, David I. Beaver, Shobhana Chelliah, and Stanley Dubinsky. 2018. Reproducible research in linguistics: A position statement on data citation and attribution in our field. *Linguistics*, 56(1):1–18.

Steven Bird and Gary Simons. 2003. Seven dimensions of portability for language documentation and description. *Language*, page 557–582.

Claire Bowern. 2015. *Linguistic fieldwork: A practical guide*. Springer. Citation Key Alias: bowern2015b, bowern2015c.

Claire Bowern, Irene Yi, and Sarah Babinski. 2022. How usable are digital collections for endangered languages? a review. Presented at the "In Case You Missed It" series of the Linguistic Society of America.

Shawn Garbett, Jeremy Stephens, Kirill Simonov, Yihui Xie, Zhuoer Dong, Hadley Wickham, Jeffrey Horner, Will Beasley, Brendan O'Connor, Gregory Warnes, Michael Quinn, and Zhian Kamvar. 2023. yaml: Methods to convert r data to yaml and back. `https://CRAN.R-project.org/package=yaml`.

Amanda Harris, Nick Thieberger, and Linda Barwick. 2015. *Research, Records and Responsibility: Introduction*. Sydney University Press. Accepted: 2015-10-07.

John Hatton, Gary Holton, Mandana Seyfeddinipur, and Nick Thieberger. 2021. Lameta [software].

Ryan E Henke and Andrea L Berez-Kroeker. 2016. A brief history of archiving in language documentation, with an annotated bibliography. *Language Documentation*, 10:47.

Felicity Meakins, Jennifer Green, and Myfany Turpin. 2018. *Understanding linguistic fieldwork*. Routledge.

David Nathan. 2010. Archives 2.0 for endangered languages: From disk space to myspace. *International Journal of Humanities and Arts Computing*, 4(1–2):111–124.

Bernard C. Perley. 2012. Zombie linguistics: Experts, endangered languages and the curse of undead voices. *Anthropological Forum*, 22(2):133–149.

R Core Team. 2017. *R: A language and environment for statistical computing*. Vienna, Austria. 00000.

Mskwaankwad Rice. 2022. Power and positionality: A case study of linguistics' relationship to indigenous peoples. *Proceedings of the Linguistic Society of America*, 7(11):5295.

Ryan Sullivant. 2020. Archival description for language documentation collections. *Language Documentation & Conservation*, 14:520–578.

Kevin Ushey, JJ Allaire, and Yuan Tang. 2023. *reticulate: Interface to 'Python'*. Https://rstudio.github.io/reticulate/, https://github.com/rstudio/reticulate.

Hadley Wickham. 2011. The Split-Apply-Combine strategy for data analysis.

Hadley Wickham, Jim Hester, Winston Chang, and Jennifer Bryan. 2022. *devtools: Tools to Make Developing R Packages Easier*. Https://devtools.r-lib.org/, https://github.com/r-lib/devtools.

P Wittenburg, H Brugman, A Russel, A Klassman, and H Sloetjes. 2006. Elan: a professional framework for multimodality research. *Proceedings of LREC 2006, Fifth International Conference on Language Resources and Evaluation.*, page 1556–1559. 00216.

Yihui Xie. 2018. xfun: Miscellaneous r functions. `https://CRAN.R-project.org/package=xfun`.

Irene Yi, Amelia Lake, Juhyae Kim, Kassandra Haakman, Jeremiah Jewell, Sarah Babinski, and Claire Bowern. 2022. Accessibility, discoverability, and functionality: An audit of and recommendations for digital language archives. *Journal of Open Humanities Data*, 8(00):10.