An Online Platform for Community-Based Language Description and Documentation

Rebecca Everson
Independent
reverson94@gmail.com

Wolf Honoré
Yale University
wolf.honore@yale.edu

Scott Grimm
University of Rochester
scott.grimm@rochester.edu

Abstract

We present two pieces of interlocking technology in development to facilitate community-based, collaborative language description and documentation: (i) a mobile app where speakers submit text, voice recordings and/or videos, and (ii) a community language portal that organizes submitted data and provides question/answer boards whereby community members can evaluate/supplement submissions.

1 Introduction

While engagement of language communities, and diverse members thereof, is crucial for adequate language documentation and description, this is often a challenging task given the finite resources of field linguists. We present a technological platform designed to accelerate and make more inclusive the process of documenting and describing languages with the goal of enabling language communities to become researchers of their own languages and curators of language-based facets of their culture.

We are currently developing two pieces of interlocking technology: (i) a mobile app whose functionality and simplicity is reminiscent of WhatsApp (which is widespread in, for instance, West Africa) through which speakers submit text, voice recordings and/or videos, and (ii) an community language portal that, for each language, organizes and displays submitted data and provides discussion and question/answer boards (in the style of Quora or Stack Exchange) where community members can evaluate, refine, or supplement submissions. Together, these permit field linguists, community educators and other stakeholders to serve in the capacity as "language community coordinator", assigning tasks that are collaboratively achieved with the community. This technology shares similarities with recent efforts such as Aikuma (Bird et al., 2014) or Kamusi (Benjamin and Radetzky, 2014); however, the focus is on developing a limited range of functionalities with an emphasis on simplicity to engage the highest numbers of community members. This in turn will accelerate the most common tasks facing a language description and/or documentation project and do so at a technological level in which all community members who are capable of using a mobile phone app may participate.

In this paper, we first exemplify the mobile app with the process of developing language resources with developing a lexicon. Section 2 contains an overview of the application, Section 3 is a discussion of the different user interfaces, and Section 4 gives the implementation details. Finally, we address extensions currently under development in Section 5.

2 Language Resource Development

Lexica and other resources are developed through the interaction between coordinators and contributors. The coordinator, who could be a linguist and/or community member, puts out queries for information and accepts submissions through a web console written in TypeScript using the React framework. Contributors use the mobile interface, built with React Native, a popular JavaScript framework for mobile development, to add words in both spoken and written form, along with pictures. The accepted submissions are used to automatically generate and update interactive online resources, such as a lexicon.

For lexicon development, the platform is able to accommodate a wide variety of scenarios: monolingual or bilingual, textual and/or audio along with possible contribution of pictures or video files. Thus, in one scenario a coordinator working on a language for which textual submissions

Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages: Vol. 1 Papers, pages 1–5,
Honolulu, Hawai'i, USA, February 26–27, 2019.

are infeasible could send requests in the form of recorded words soliciting "folk definitions" of those words (Casagrande and Hale, 1967; Laughren and Nash, 1983), that is, the speakers themselves supply the definition of a word, typically producing more words that can be requested by the coordinator. Alternately, semantic domains may serve as the basis of elicitation in the style of Albright and Hatton (2008) or in relation to resources such as List et al. (2016). Built into our approach is the ability to record language variation: For lexicon entries, multiple definitions and forms can be provided by different speakers, which can then be commented and/or voted on, until a definition (or definitions) is generally accepted and variation is properly recorded.

Analogous processes allow speakers to contribute varieties of language-based cultural content: folktales, oral histories, proverbs, songs, videos explaining cultural practices and daily tasks (cooking, sewing, building houses, etc.). Accordingly, this method may be used to develop (i) data repositories and resources for researchers in linguistics and allied fields, especially those touching on studies in language and culture, and (ii) educational materials and other materials that may benefit the community.

3 User Interface

For the purposes of this section, we focus on the task of developing a lexicon on the basis of a predetermined wordlist, such as the SIL Comparative African Word List (Snider and Roberts, 2004), a common scenario for a fieldworker working on an under-described language for which the speakers also speaks an administrative language such as English or French.

Users of the application fall into three categories: coordinators, contributors, and consumers. A coordinator can be a member of the language community, such as an elder or group of elders, or she might be a field linguist working within the community. Coordinators handle adding new words to translate, assigning words to contributors, and accepting or rejecting submitted translations. Coordinators also have the ability to assign contributors and words to subgroups. This is useful if there are specialized vocabularies specific to only a portion of the community, for example, hunters who use a different vocabulary while hunting. A contributor is a community member who has been chosen by the coordinators to complete the translation work. Finally, a consumer is anyone who has access to the community language portal that is created from the submissions accepted by the coordinators.

We see the development of the community language portal and the presentation of speakers sharing their language as key for motivating continued contributions. We expect that the varieties of language-based cultural content speakers contribute as part of the documentation activities, e.g., folk tales or oral histories, will be a key motivator for community members to use and contribute to the platform. We provide functionality for consumers to also become contributors. For instance, next to contributed videos, a button allows consumers to contribute their own video. Content so contributed will not be directly accepted to the database, but will require approval from the coordinator, so as to provide a check on inappropriate content.

The following subsections give sample user stories for each type of user. As will be detailed in Section 4, users will sign in with a Google account, which provides a straightforward solution for user authentication. In the following user stories we use English and Konni [kma] as our example languages, though the application can work for any pairing.

3.1 Contributor

A new contributor joins the language description and documentation effort. They belong to a Ghanaian community that speaks English and Konni, which is the language they want to describe and document. They open the mobile application and sign in with their Google account credentials. Upon the initial sign in, the contributor must provide responses to secure consent in accordance with an IRB protocol. Then the contributor is presented with a demographic survey. Once successfully logged in, they see the home screen, which displays their assignments, e.g., a list of semantic prompts or words in English (see Figure 1). They select a word from the list by tapping it. They are then taken to a form with fields for the translation of the word in Konni, a sample sentence using the word in English, and a translation of the sentence in Konni. There are also two fields for audio recordings of the word and sentence in Konni. When the user selects these fields they see a screen

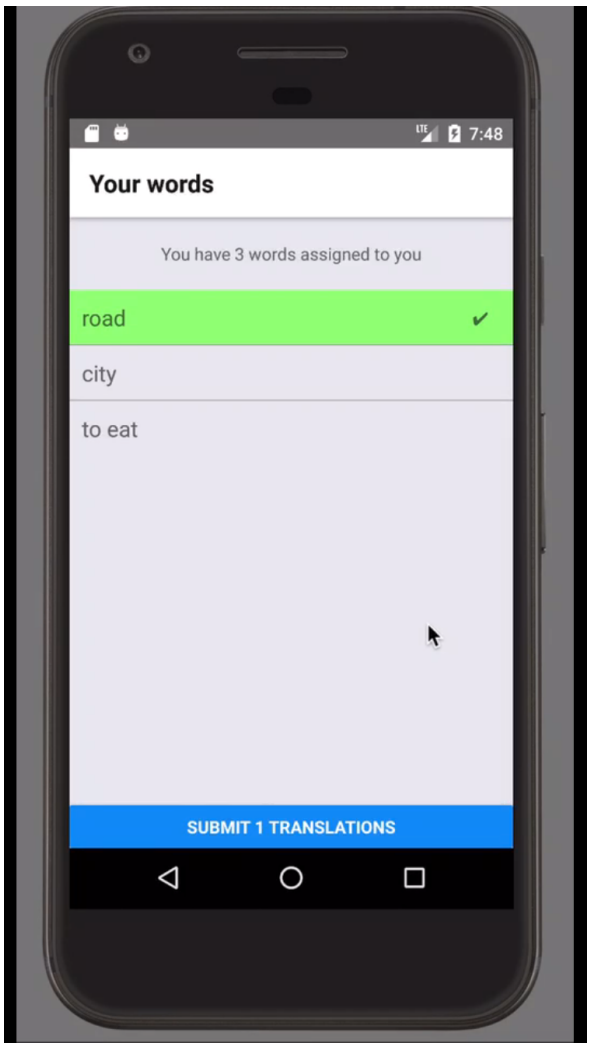

Figure 1: Application home page.

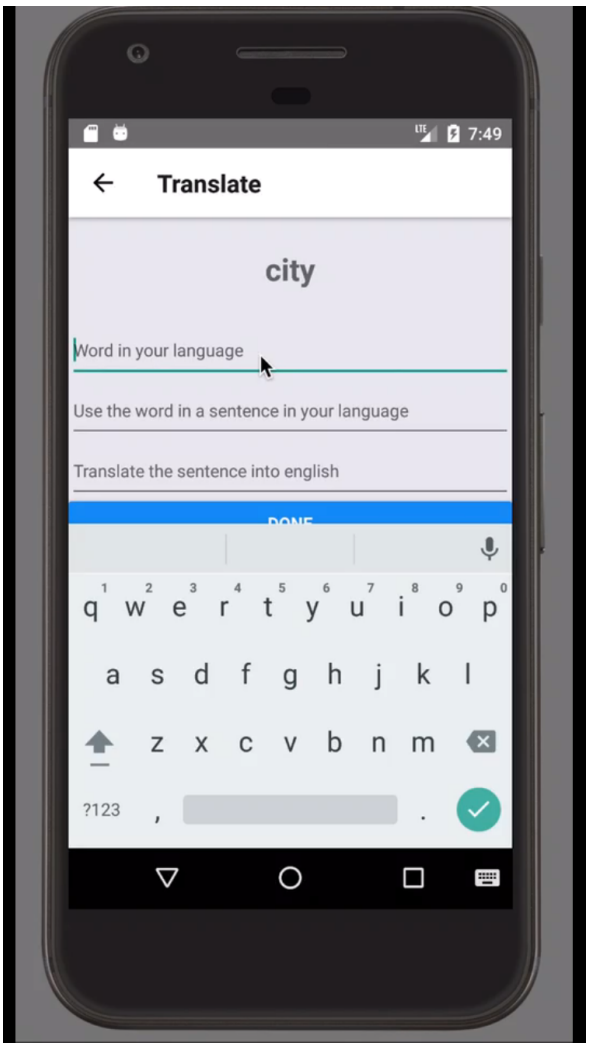

Figure 2: Word translation page.

with *Start* and *Stop* buttons. Pressing *Start* begins recording using the phone's microphone and *Stop* ends the recording and saves the audio as a WAV file. When they have filled in all of the fields, they press the *Submit* button at the bottom of the screen and are taken back to the home screen. Now the word that they just translated is green and has a check-mark next to it (see Figure 2). They now also see a button at the bottom of the home screen that reads *Submit 1 translation(s)*. When they are ready to submit their completed translations, they press that button. If their phone is currently connected to the Internet, the translated words will be removed from the list and new words will appear. If not, they can continue translating or wait to enter an area with WiFi.

3.2 Coordinator

A coordinator wants to review some translations and assign more words to contributors. He starts by opening a web browser and navigating to his documentation project's web page. After logging in with his Google account credentials he sees that a contributor has just submitted three new translations that need to be reviewed. The coordinator reviews all of the translations and decides that two of them are ready for publishing, but one of them needs a better example sentence. He rejects the translation with a note explaining why it was rejected, and the word is put back into circulation automatically by the system. The two accepted words will no longer be added to users' word lists unless manually added back into the database.

3.3 Consumer

A consumer who visits the Community Language Portal navigates to a part of the website providing a picture gallery of everyday and cultural artifacts. As she looks through the pictures and their captions, she notices that the word for *water pitcher* has a different form than is used in her village. She clicks on the 'Contribute' button, and is able to either leave a comment about the caption or go through the sign-up process to become a contributor. The coordinator reviews her submission, and, if appropriate, adds her to the contributors, later sending out more invitations to contribute.

4 Implementation Details

Users of the application only see information relevant to their current task, but all user and language data is stored and managed in a remote database. The application communicates with a CherryPy web server, which acts as the glue between these two components. For example, consider the Konni contributor trying to submit some translations. Before doing anything, the user must first sign in. To avoid having to manage passwords we require users to connect to the application using a Google account. Then the application communicates with Google's Authentication Server using the OAuth 2.0 protocol to retrieve an authentication token that uniquely identifies the user. When the contributor presses *Submit*, the phone will first store the translation information locally and wait until it enters an area with a strong Internet connection. Once connected, it then sends an HTTP request to the web server containing the translations to be submitted (multiple translations are batched together for efficiency), as well as the authentication token. Upon receiving the request, the server uses the token to verify that the user exists and has the proper permissions. It then adds the translations to the database and queries it for a new set of untranslated words. Finally the server responds to the application indicating that the submission was successful and containing the new words.

At the lowest level of the application stack is a MySQL database that is responsible for storing user information and translations. It consists of three tables: `words` for the words that are to be translated, `users` for all the application users, and `translations` for all submitted translations, both reviewed and unreviewed. An entry in `words` contains the word itself, as well as other grammatical information, such as part-of-speech. The `users` table contains users' names, their Google authentication tokens, their roles (contributor or coordinator), and the maximum number of words that can be assigned to them. Each row in `translations` consists of the word translated to the target language, a sentence containing the word in the source language, the same sentence translated, paths to audio files containing recordings of the word and sentence, and a flag indicating whether the translation has been accepted.

This is a natural division of the data that allows the tables to grow independently of one another (e.g. adding a new user only affects the `users` table). However, we often want to make queries that depend on information from multiple tables, such as searching for words with no accepted translations that are assigned to fewer than three users. To facilitate these searches we also introduce links between the tables. A word can have multiple translations, but each translation corresponds to exactly one word, so `words` and `translations` have a 1-many relation. Similarly, because a user may have many submitted translations, and each translation was submitted by one user, `users` and `translations` also have a 1-many relation. On the other hand, a word can be assigned to several users, and a user may have multiple assigned words, so `words` and `users` have a many-many relation.

5 Current Developments

With this groundwork laid on the application, we are expanding other aspects of the project. Since one of our primary goals is to engage community members, we are pursuing more ways for them to engage with the software and data. To that end, we are designing a discussion board for raising questions about accepted materials, asking for clarification on words, debates, polls, and so forth. Anyone from the community will be able to use this software. In short, we aim to leverage successful examples of online community building to further language description and documentation.

It is possible that contributors use slightly different lexica within the community. For example, in a community that has a designated group of hunters, the hunters might use different words in the field that community members who stay in the village most of the time don't know. In this example, a coordinator might want to gather lexical data

from both groups, so they should be able to mark which users belong to which sub-communities in the database. At the moment, we have back-end functionality for this sub-grouping of words and users, but no way for a coordinator to interact with this feature from the application. In the meantime, words are assigned to users automatically.

Having a way for a coordinator to assign words to specific users will also be an important feature. It is very likely that contributors will sometimes be working in areas with a lot of background noise, and not everyone will have a phone that can record high-quality audio. Giving coordinators the ability to reassign words to users who they know can record with better sound quality will ensure high-quality data, which can then be used later in linguistic analysis.

References

Eric Albright and John Hatton. 2008. Wesay: A tool for engaging native speakers in dictionary building. *Documenting and revitalizing Austronesian languages*, (1):189.

Martin Benjamin and Paula Radetzky. 2014. Small languages, big data: Multilingual computational tools and techniques for the lexicography of endangered languages. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 15–23.

Steven Bird, Florian R Hanke, Oliver Adams, and Haejoong Lee. 2014. Aikuma: A mobile app for collaborative language documentation. In *Proceedings of the 2014 Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 1–5.

J.B. Casagrande and K.L. Hale. 1967. Semantic Relationships in Papago Folk-Definitions. In *Studies in Southwestern Ethnolinguistics*, pages 165–193. Mouton, The Hague.

M. Laughren and D. Nash. 1983. Warlpiri dictionary project: Aims, method, organization and problems of definition. *Papers in Australian Linguistics*, 15(1):109–133.

Johann-Mattis List, Michael Cysouw, and Robert Forkel. 2016. Concepticon: A resource for the linking of concept lists. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation, May 23-28, 2016, Portorož, Slovenia*, pages 2393–2400.

Keith Snider and James Roberts. 2004. Sil comparative african wordlist (silcawl). *JWAL*, 31(2):73–122.