Resourcing Small Indigenous Languages in the Field

Antria Panayiotou[1], Francis Saa-Dittoh[2], and Anna Bon[1]

[1] Vrije Universiteit Amsterdam, The Netherlands
antria_pan@outlook.com, a.bon@vu.nl
[2] University for Development Studies, Tamale, Ghana
franditt@gmail.com

Abstract. Small indigenous languages of the world are often spoken in regions where literacy rates are low and access to digital services is limited. Whereas voice information services could be useful for low literate speakers of these languages, written and audio data from these languages are hardly available. This paper presents a method for resourcing indigenous languages, with the aim to build targeted voice services based on an automatic speech recognition system. The proposed method for data collection has been co-developed with native speakers of an indigenous language, *dagbanli* and validated in the field, through a case study in northern Ghana.

Keywords: Under-resourced indigenous languages, small data, low-resource environments, user-centered design, agile development, ICT4D

1 The importance of resourcing small indigenous languages

Indigenous languages are often spoken in regions where educational systems are lacking, and consequently, literacy rates are low. In these regions, generally, purchasing power is low and people are less connected to digital services. In many countries in the Global South, that have been colonized in the 18$^{\text{th}}$ and 19$^{\text{th}}$ centuries, the colonizer's language has been imposed as *lingua franca*. In Africa, English, French and Portuguese have been introduced and have rapidly spread through newly introduced, European educational systems. Adoption of the colonizer's language had many advantages, as a channel for communication and trade. However, this pushed away many indigenous languages, of which many have disappeared or are at the brink of extinction [1].

There is a growing interest in the conservation of culture and indigenous knowledge and languages. Digitization has the potential to preserve these languages and connect large groups of still "unconnected" people. However, costs of voice services are generally high. Digital data such as text or audio from indigenous languages are not widely available.

In this paper we present a detailed case study of designing a method for data collection for small indigenous languages. By co-designing, testing, improving, it

with native speakers of *dagbanli* language in northern Ghana, the method has been validated. In this paper we aim to generalize our findings and draw lessons how to resource small indigenous languages in general.

2 Previous work

Various studies have investigated how online crowd-sourcing is used for data collection for automatic speech recognition.

Toshniwal et al. (2018) [3] present a method for multilingual speech recognition involving training of a singular end-to-end model on multiple languages. Compared to conventional models that are trained separately for each language, this method shows comparable/superior performance. The authors demonstrate that this method is computationally economical, as the model can be trained simultaneously on multiple languages without requiring additional resources.

An open-source platform named Woefsela, to collect speech data in low-resource environments was presented by De Vries et al. [4]. The platform is designed to operate on low-end Android devices and includes real-time quality control to ensure that the acquired data is of sufficient quality for training speech recognition models. The study describes how Woefzela was utilised to capture speech data for some South African languages to train speech recognition models.

According to a study by the Interaction Design Foundation [3], prototypes play a crucial role in the design process, allowing designers to explore and test various ideas, collect feedback, and make informed decisions about the design's direction. Also, according to the National Institute of Standards and Technology, prototypes are a powerful tool for evaluating and improving designs because they allow designers to find and fix problems early in the design process before it becomes more costly to do so[4].

Due to limited internet connectivity, crowd-sourcing platforms do not reach many people in less connected areas. To mitigate this problem, Stan et al. [2] presented a light-weight method to resource small speech data using a Machine learning (ML) algorithm. A small corpus speech recognition system was built as a proof-of-concept.

In this paper, we intend to validate Stan's method [2] using speech data from an indigenous language. To accomplish this, a system for field-based data collection must be designed and built. The collected speech data will be used at a later stage to train an Artificial intelligence (AI) model on a comprehensive and diverse dataset. Sufficient variation in pronunciation and intonation of the collected speech are needed to improvement of the automated speech recognition model's accuracy.

3 Problem statement

Resourcing indigenous languages presents a number of obstacles. In regions where indigenous languages are spoken, the deployment of digital technologies is

[3]https://www.interaction-design.org/literature/topics/prototyping

[4]https://www.nist.gov/publications/prototyping-new-products-systems-and-processes

frequently hampered by numerous infrastructural constraints, for example, limited internet. Data must be collected with users in the field. Meanwhile, there is typically a language and cultural gap between the envisaged users and the ICT system's technical developers. Given the complexity of the local context, the focus of this paper is on determining the optimal method for designing and constructing a data collection system to produce a digitised language corpus.

4 Methodology and approach

The general approach to designing digital technologies in low resource environments is to co-design with local users in a living lab setting and learn from the experience and the feedback. The first action is a stakeholder analysis, including needs assessment. Next, a use case and requirements analysis is carried out. We choose agile development with short iterative cycles of rapid prototyping [5]. This breaks the development process into small, manageable stages of one or a few days, incorporating user feedback at every stage, facilitating the creation of a user-friendly interface, that meets the users needs [6] [7].

For the design, we adhere to the guidelines from the 10 Nielsen rules and NASA design, developed in the 1990s by Jacob Nielsen [9], and of the Human Interaction Design Protocol by the "NASA's International Space Station Program at Johnson Space Center in Houston, Texas"[10]. These are guidelines to design for safety, accessibility and adaptability. This includes communicating with users in language and concepts that are familiar for them, while giving them autonomy and control. It stresses the importance of simplicity and clarity of the design, and of consistency and predictability. This results in an intuitive and enhanced experience for users with different levels of skills.

The following two sections describe a case study in Ghana, in which a data collection application for resourcing of the small indigenous language *dagbanli* was developed. This was done iteratively, in a living lab setting, in which the users were able to test the application in the local context, at every stage of its development. From the findings of this field-based case study we draw lessons how to resource small indigenous languages.

5 Tiballi, a field-based case study in Ghana

This research is carried out in northern Ghana, with the objective to develop ASR-based voice services to provide weather and rainfall information to rural communities in their own native language. Developing a data collection application to generate a speech corpus is the first activity in this research project, dubbed Tiballi, which means "our language" in local *dagbanli*.

The data collection tool targets online communities that speak the chosen language and are already involved in community-centered work. The launch of the project involves a workshop that brings together stakeholders in the field: farmers from the communities, AI, linguistic and ICT4D experts from various organizations. These stakeholders form a group of knowledgeable individuals who

will provide data and facilitate the spread of crowd-sourcing at the appropriate time. The process also employs individuals with experience in data collection. They are compensated to visit the communities and collect voice fragments from native speakers. Data collection starts in the selected Nyankpala and Tingoli, communities, in the Tolon District, and may be expanded to other regions.

5.1 Use case analysis

The project starts with a use case analysis, to find out the objectives and requirements for an app to collect utterances of dagbanli language. The next question is how the data collection app will be used and who the users are. Two of the most common use cases that are done though the app are displayed in Figure 1:

Use case 1: User perspective (dagbanli speaker)

User persona 1: A dagbanli speaker who wants to contribute to the community by recording their voice and helping to create a knowledge base for a ML program.

Use case 2: Developer perspective

User persona 2: The developer of the data collection app wants to monitor the contributions and performance of the app.

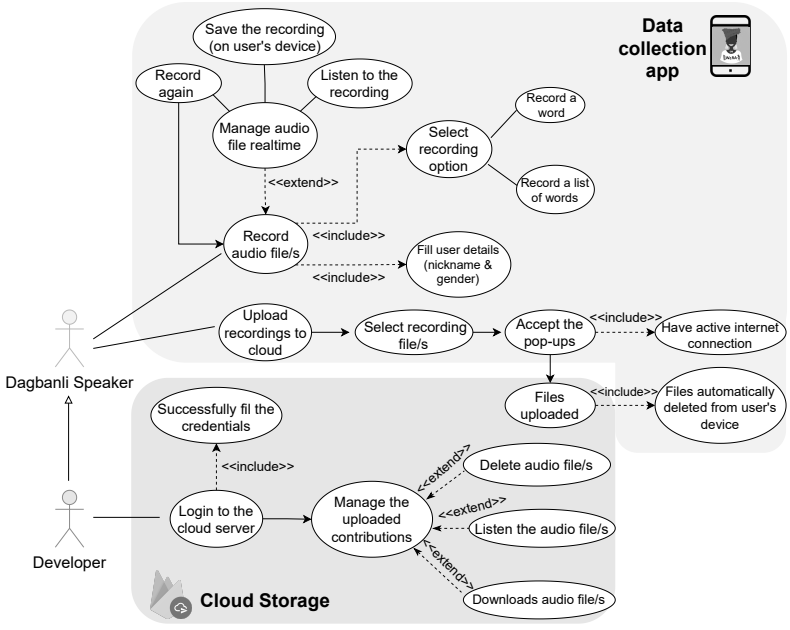

Fig. 1. Use case scenarios for the personas.

5.2 Stakeholder analysis

Identifying stakeholders and their specific needs with respect to the app is the 1st stage of the analysis. This project's stakeholders include app end-users, developers and the originators of the project concept. Secondly, given the poor local (digital) infrastructure, and the technical requirements for data collection, a digital platform should be available which is scalable, maintainable, and secure for the collection of a language corpus of sufficient quality and size for voice data for model training. Thirdly, attention must be given to the human computer interaction aspects of the app. The app's functional and non-functional requirements are described in detail in the subsection 6.1.

The 1st round of user feedback is mapped using the MoSCoW method, a technique for ranking importance of user requirements against ease/complexity/cost of development, and prioritizing which *must* be in the app, which *should* be, *could* be or *would not* be built in the app [8].

6 Designing a data collection app

6.1 Initial requirements

The most basic technical and general requirements for the data collection app are specified and listed as (i) functional and (ii) non-functional requirements. The functional requirements:

- An audio recording's data storage must be arranged.
- Quality of the recorded speech must meet the machine leaning models.
- If the recorded audio quality is below quality, there must be an option for quality enhancement.
- A human quality control function must be in place to ensure the accuracy and quality of user-submitted transcriptions of the utterances.

Non-functional requirements relate in general to the performance, compatibility, and usability, including performance, security, and usability of an app. The following are collected for the envisaged app:

- The app must be able to quickly and efficiently process a certain volume of voice recordings.
- The app must function on multiple platforms: Smartphone and Desktop computers, to facilitate user accessibility in the field and at the office.
- An intuitive human-computer interface for recording and submitting voice data is required, that meet s the local users' level of computer literacy.
- The audio must be of sufficient quality and volume, that in a subsequent phase, a specified ML model must be able to accurately recognise and transcribe the recorded utterances for the given language.

6.2 User requirements refined - 1st iteration

To ensure that the project's user requirements are precisely captured and refined during the 1st iteration of user tests, prototyping techniques are used, such as the creation of a video and a website prototype, to communicate the project vision with all stakeholders. This process yields feedback that assists updating and refining the end-users' requirements. Rapid prototyping is a vital component of the design process, ensuring that design is rapidly adapted to the emerging requirements and expectations of test users.

In the prototyping stage, the developer employes resources such as proto.io[5] to construct interactive and operational prototypes. Proto.io provides a diverse range of functionalities such as audio recording, cloud-based file uploading, and cautionary pop-up notifications that enhance the design's realism and facilitate users in evaluating the design as if it were an actual product.

The interactive feature facilitates the evaluation of design functionality and the identification of potential impediments, enabling designers to make informed decisions regarding the design's trajectory and solicit feedback from relevant parties. The user emphasises that Proto.io's interface is intuitive and suitable for designers of varying skill levels. In addition, the tool is renowned for its capacity to improve design quality and streamline the prototyping process.

In addition, a pre-recorded video prototype[6] of the design is generated. This elucidates functionalities and characteristics of the design in a non-interactive static format. Despite limitations in comparison to real-time prototypes, this approach is commonly employed in conjunction with other prototyping techniques to offer the users comprehension of the design and its potential impact.

The 1st iteration took place in Ghana, during a field visit to the local community of Tingoli, in February 2023. Deliberations led to the decision to develop two prototypes. Following the workshop, several modifications were implemented, including revisions to the text to enhance clarity and redesigning. Moreover, the data categories for the collected utterances were, for the 1st stage deployment, reduced to two. The two categories under consideration comprise numerical values 0 to 10 and a category for collection of "Yes" and "No" in dagbanli language. Additional word categories for collecting for example for weather and rainfall (rain, storm, drizzle) or relevant local crops (maize, rice, groundnuts) are planned for a later phase of data collection.

6.3 User feedback - 2nd iteration

Upon implementation of the new requirements, the initial iteration of the mobile version of the app was deployed and shown to the users in Ghana. The results of the 1st cycle's requirements are shown below:

[5] https://proto.io/

[6] https://www.youtube.com/watch?v=U6P6FmC2_4s

The main screens from the data collection app: [7] To record one single word, the user selects the "one word" option in the *ratio category* as displayed in Figure 2. Once selected, the user can view available words, choose a desired word, and record it. "Record again", "Save", and "Play", to save or re-record it. As the user enters a category of words, additional options such as "Skip the word" and "Save and continue" will become accessible. By selecting the "Skip the word" icon, users can bypass unfamiliar terms. Additionally, users can save their progress by selecting the "Save and Continue" button, which displays the next word to be recorded. Once the last word in a category has been recorded, the "Finish and Submit" option becomes available.

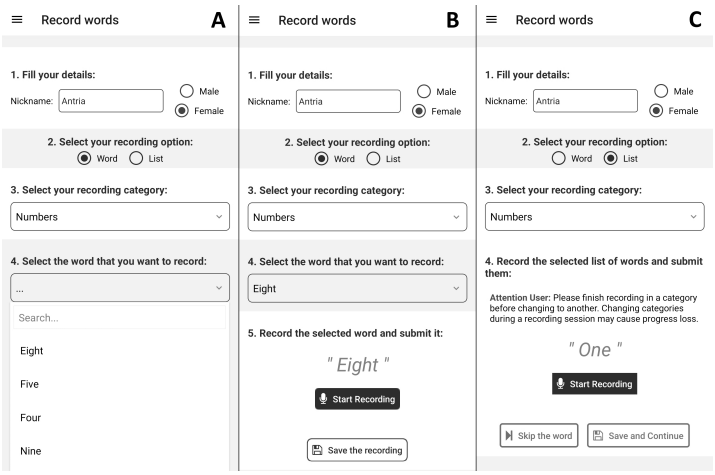

Fig. 2. Screenshots from the mobile app during the recording process [A: Selecting the recording word, B: Available "word" options, C: Available "list" options].

Before uploading, it is necessary to verify the stability of the Internet connection. An unstable connection prompts the user to retry the upload, whereas a stable connection initiates the upload process automatically, see Figure 3. A loading indicator indicates upload status and progression. A message is displayed when the upload status reaches 100 percent. The app requires access to the device's microphone and storage for respectively audio capture and local storage of data.

The following recommendations were made during 1st user evaluation:

- incorporating a "simplified" mode to facilitate independent app navigation.
- providing the option to display the interface in the local language improves accessibility and usability for individuals who are fluent in the local tongue.
- recommendations were given to enhance the procedure of documenting and classifying vocabulary and expressions.
- restricting the frequency of pop-up displays during the uploading process, for a more streamlined and effortless process was proposed.

[7] The source code can be made available upon request: https://github.com/AntriaPan/TIBaLLi-project-voice-services.git

– substituting the "category" parameter with "list" to avoid user confusion..
– restoring the previously selected category and word option after word recording, and categorising contributions by recording time or speaker.
– incorporation of both full-sentence recordings and single-word recordings.
– classification of recordings based on variables such as the speaker's gender, age, or ambient noise. The aforementioned improvements have the potential to offer users a more thorough and genuine educational encounter.
– load words and categories to enhance efficiency and scalability of the process of updating and managing the app's content.

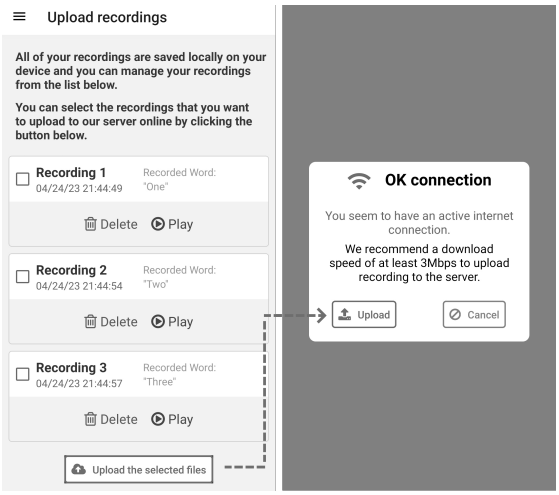

Fig. 3. Screenshots from the mobile app from the uploading process.

The aforementioned requirements and recommendations focus on functionality, accessibility and user engagement. An example of a proposed requirement that was dismissed was the option to record data of the contributing speakers. The project team dismissed this option in the app, because of increased complexity, when dealing with personal information.

Another requirement entailed consolidating all recording functionality onto a single page, saved with one single button click. This requirement was dismissed as data could get lost during recordings, in case of intermittent internet connectivity. So, partitioning the actions on different screens remained the preferred design option.

The team debated multilingual functionality, but decided to include English translations of basic terms such as "yes" and "no" and numerical values in brackets next to recorded words, or vice versa. This makes it easier for proficient readers to recognise words based on their phonetic properties.

Ultimately, the team implemented minor modifications to the app, including reduction of pop-up frequency during the uploading procedure, substitution of the "category" feature with "list," and alteration of the "word" option's return state following each recording, with the aim of enhancing user-friendliness.

These modifications are considered significant to enhance user convenience and expediency, while avoiding potential ambiguity.

6.4 Prioritizing requirements using MoSCow

When designing a system, it is necessary to balance the relevance of user requirements against the ease and cost of incorporating them in system. We used the MoSCoW method to map two dimensions, value for the user against cost of implementation. After deliberation the project team balanced the estimated relative value of each requirement, as show in the diagram on Figure 4.

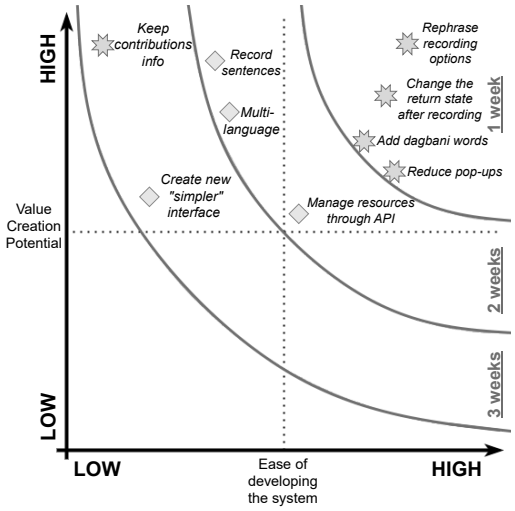

Fig. 4. Prioritization of mobile app feedback using the MoSCoW method [Stars: selected implemented feedback; Diamonds: feedback not selected].

6.5 User feedback - 3rd and final iteration

During the 3rd iteration the mobile app was completely recreated and submitted for evaluation, to the satisfaction of the users in Ghana. A last comment entailed the incorporation of a text field for input of name and gender of the data collecting person. This feature was deemed indispensable for the purpose of audio file analysis in a subsequent phase. A salient facet was whether they would furnish their complete name or just a nickname. Following discussion, the decision was taken for a nickname, as many people share the same name, particularly in this region of Ghana, where the app is being utilised. This resolution will facilitate expansion of the app to broaden its user base.

7 Results and conclusion

This study resulted in (i) a data collection app (ii) a general method to resource indigenous languages. Despite the contextual limitations and specific requirements of this project, we can generalize the following findings :

- Resourcing an indigenous language must be done in close collaboration with native speakers of the targeted language.
- Context analysis and stakeholder analysis are essential and need to be done at the start of the project.
- To bridge the communication gap between developer and user(s), field visits, test sessions and frequent meetings are required.
- Real-life demos followed by rapid prototyping and frequent user-tests are appropriate for elicitation of requirements.
- User requirements are collected and documented, using a method of prioritization of value against costs/ease of implementation (MoSCow).
- Context-awareness of the conditions and limitations are key success factors in designing for low-resource environments.

8 Acknowledgement

We thank André Baart and Gideon Amakama Ali for their contributions to this study. We are thankful to the Internet Society, for the research grant that was provided to carry out the Tiballi project in northern Ghana.

References

1. Ndebele, H.: Promoting indigenous African languages through information and communication technology localisation: A language management approach. 2020, Alternation Special Issue, 13, 102-127.
2. Stan, G.V.: "Small" language limited-vocabulary automatic speech recognition using ML. Web alliance for Regreening in Africa, 2021.
3. Toshniwal, S., Sainath, T.N., Weiss, R.J., Li, B., Moreno, P., Weinstein, E., Rao, K.: Multilingual Speech Recognition With A Single End-To-End Model, 2018. arXiv preprint arXiv:1808.02610.
4. De Vries, N.J., Badenhorst, J., Davel, M.H., Barnard, E., De Waal, A.: Woefzela-an open-source platform for ASR data collection in the developing world. Conference paper, 2011.
5. Bon, A. ICT4D 3.0, - An adaptive, user-centered approach to innovation for development. 2016, March, doi: 10.13140/RG.2.1.3260.5687.
6. Cao, L., Mohan, K., Xu, P., Ramesh, B. A framework for adapting agile development methodologies. European Journal of Information Systems, 2009, 18(4), 332-343. doi: 10.1057/ejis.2009.28.
7. Salah, D., Paige, R. F., & Cairns, P. A systematic literature review for agile development processes and user centred design integration. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, 2014, (pp. 1-10).
8. Ahmad, K. S., Ahmad, N., Tahir, H., Khan, S. Fuzzy MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements. In 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2017, (pp. 433-437). IEEE.
9. Nielsen, Jakob and Molich, Rolf, Heuristic Evaluation of User Interfaces, 1990, Association for Computing Machinery, 249-256, doi:10.1145/97243.97281
10. Jennifer Boyer, Human Integration Design Processes (HIDP), 2014