

LanguageARC – a tutorial

Christopher Cieri, James Fiumara

University of Pennsylvania, Linguistic Data Consortium

3600 Market Street, Philadelphia, PA 19104 USA

{ccieri, jfiumara}@ldc.upenn.edu

Abstract

LanguageARC is a portal that offers citizen linguists opportunities to contribute to language related research. It also provides researchers with infrastructure for easily creating data collection and annotation tasks on the portal and potentially connecting with contributors. This document describes LanguageARC's main features and operation for researchers interested in creating new projects and or using the resulting data.

Keywords: language resources, crowd-sourcing, citizen linguistics

1. Introduction

LanguageARC is a portal that connects researchers to citizen linguists who may be interested in contributing to research projects (Figure 1). It was created as part of the NIEUW project which investigates novel incentives in the elicitation of language related data as a way to fill the gaps in available language resources left by other approaches.

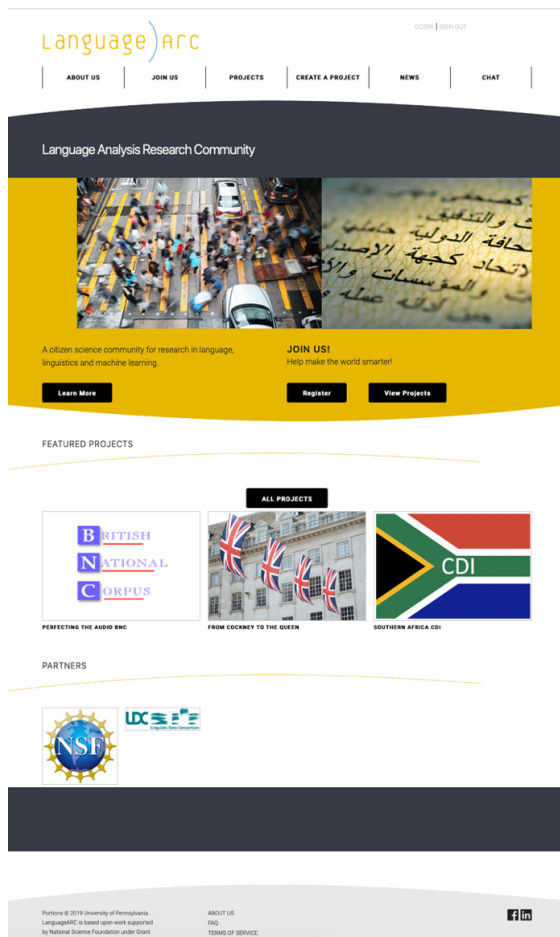


Figure 1: LanguageARC Home Page

Other NIEUW outcomes include the language games portal LingoBoingo.org and the language identification game NameThatLanguage.org which offer the incentives of entertainment, competition and opportunities to learn in exchange for language data. In contrast, LanguageARC

offers members of the public interested in language (citizen linguists) opportunities to learn about and make direct contributions to research on language and to join groups of like-minded contributors.

LanguageARC includes a project builder that vastly simplifies the steps required to create and deploy a cluster of related web pages that collect data and annotation. Two design goals are that: 1) tasks should be simple and short enough to be completed by citizen linguists, for example, while commuting, on a work break, waiting for an order in a restaurant, etc. and 2) that researchers should be able to implement new tasks in less than one hour given a design and data in the appropriate format. These design goals are intended to lower the barriers to participation for both researchers and citizen linguists.

2. Terminology

LanguageARC's principal organizing scheme is that the portal hosts multiple *projects*, each of which contains one or more *tasks*, each of which iterates over one or more *items*. A *project* is a set of tasks organized by a research team to support a specific research goal. LanguageARC tasks are organized by project – rather than, for example, by language, activity type or application – to give research teams the opportunity to describe their work in a way that attracts citizen linguist *contributors*. To appeal to contributors, a project has a compelling *project image*, *title*, *call to action* and *description*. Each project is represented by a *card* on LanguageARC's multi-page grid of all projects (see Figure 2). The card displays the project's image, title and call to action. Clicking any card takes the user to that project's *main page*.

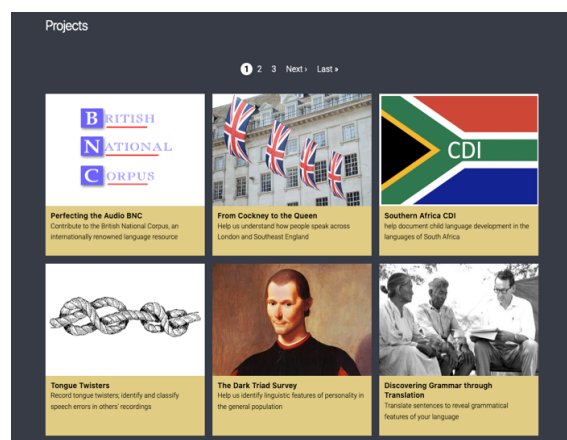


Figure 2: Project grid (partial)

The project main page (Figure 4) repeats the project title, call to action and image but also adds a *description*, optional *partner badges* and optional links to *News*, *Chat* and *Research Team* pages. Currently, there is no blog implemented within the portal but projects that have their own external blog or web pages can use the News link to connect contributors to those. LanguageARC does have its own discussion groups accessible via the Chat link. The project main page also contains a large button that reads *Start Now* for new contributors and *Continue* for returning contributors.



Figure 4: One project's main page (partial)

Every project must have at least one task but projects can have many more than one. If a project has multiple tasks, the Start Now/Continue button takes the contributor to the *task list* (Figure 5); otherwise it starts the single task immediately. The task list page inherits any Research Team, News and Chat links from the project main page but add an image, title, call to action and Start/Continue button for each task within the project. Clicking the Start/Continue button for any task takes the contributors to the tasks tool page.

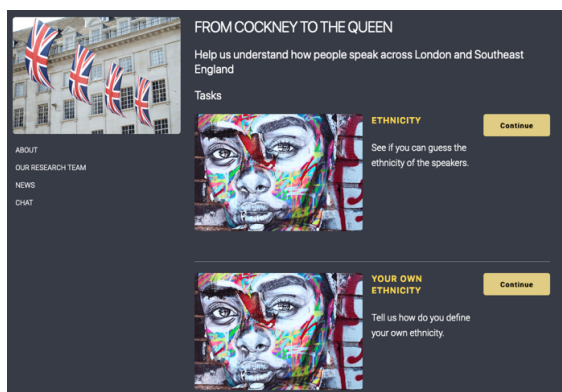


Figure 5: Task list for a project with multiple tasks (partial)

Each task has one and only one *tool* page (see Figure 3). This is where most of the work is done. The tool is built from widgets or controls, customized for the task, that allow the contributor to play audio or video, read text or view images and then contribute language data by typing or recording themselves speaking responses or by clicking buttons. Each tool page can include optional links to a *tutorial* and *reference guide*. Each task performs the same action over one or more items in a data set. A *data set* is

defined as a *manifest* that enumerates a set of items by providing identifiers for each item as well as item specific texts, media files or both. Media files can be text, audio, image or video.

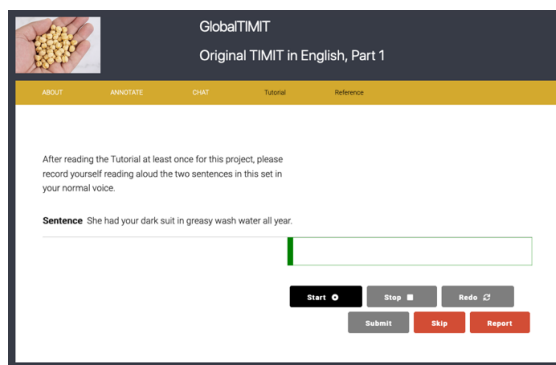


Figure 3: Tool page (partial)

3. Preparing a Project

Before beginning implementation, *project designers* consider their research goals and the subset of tasks citizen linguists could do. Citizen scientists contributing to other portals such as *Zooniverse* have demonstrated their willingness to learn complex tasks and ability to complete them with high quality. Nevertheless, it remains the case that human performance is better for straightforward tasks with clear instructions that require contributors to make one kind of decision at a time. For example, if the research required both collecting transcripts and judgments about the pronunciation of audio segments, the project designer would divide that effort into two tasks. LanguageARC reflects this approach by holding the tool and instructions constant across all items within a task.

Once the project designer has defined collection and annotation, the next step is to segment any media into the units over which decisions are to be made. For example if the research goal were to transcribe conversations, the project designer would first divide the conversation into e.g. pause groups (of 4 to 8 seconds duration) which would likely require 1-2 minutes to transcribe, about the right length for a single item.

With tasks defined and media segmented, the next step is to create a manifest. A manifest is a text file of all of the items to be collected or annotated, with each item on its own line and columns separated by tab characters. Those items will be presented to citizen linguists one at a time in the tool. The manifest must always have an identifier for each item and either one or two *item specific texts* or a media file name or both. Thus a minimal manifest has two columns and a maximal one has four.

Item identifiers are required as they link the items in the manifest to the citizen linguist contributions in the automatically generated reports. The identifier can be any string of characters including a second copy of the media file name. Most projects to date have used a simple numeric counter.

Manifest files can be built from a spreadsheet that has each item in a row with the ID, item specific text and media file names in spreadsheets columns by saving the spreadsheet in the TSV (tab separated values) format. A project designer could also create a manifest directly using

a plain text editor (not a Word Processor) by placing each item on its own line with tab characters separating the ID, item specific text and media file names. columns. In the latter case, project designers should assure that the text editor is inserting actual tabs and not sequences of space characters.

With the manifest complete, the next consideration is training. LanguageARC project designers can associate a separate tutorial and reference guide with each task. In projects created so far, the tutorial introduces the task, provides any background information needed and describes the decision or other contribution to be made and perhaps repeated. The reference guides normally include screenshots of the interface with explanations, exemplars of annotation categories, definitions of terminology and acknowledgments, e.g. to people who have provided media used in building the task.

To expedite implementation, project designers gather media files to annotate and any supplemental media used in the training materials, create the manifest file and write instructions in advance of using the project builder.

Before a researcher can create LanguageARC projects, they must be given credentials as a project designer which they can request from the authors. A researcher logs into an authorized LanguageARC account will see a *Create Project* button in the main menu. Project designers can create new projects, multiple tasks within those projects and datasets for use by those tasks. They can also invite collaborators to join their projects as *task designers*, with power to edit specific tasks, or as *other contributors*. Within a **task**, task designers have all the power of project designers but cannot change **project** details or create new tasks. To avoid being tedious, we will use “project designer” below but the reader should interpret this to include “task designers” when we are discussing creating or editing task elements. *Other contributors* refers to the subset of LanguageARC contributors who have been invited, and thus have access, to a specific project or task before it is published. Finally, project designers can run reports of all contributions made to their tasks. After the project designer has tested a project and its tasks and believes it ready for public access, they use the project builder to send a request to LanguageARC *portal managers* that the project be published. Portal managers review the project to assure that it is appropriate in goals and content and that no sensitive personally identifiable information is requested. Once published, the project is available to any member of the public who creates a LanguageARC account.

4. Creating a Project

As above, preparing material in advance expedites the implementation of a LanguageARC project. Projects require an internal name, title, call to action, image and description. Not required but strongly suggested are the page about the research team and partner badges which may help attract contributors. Projects can optionally include links to an external blog or website and any of four forums associated with the project.

The *internal name* of the project is what will appear in the project builder. It need only be globally unique (not used elsewhere in LanguageARC) and memorable to the designer. The project *title* is displayed prominently on the project main page and on the project

card that appears in the grid. This title must be globally unique and should be both descriptive and attractive to potential collaborators. The *call to action*, also called the subtitle in the project builder, is normally a short phrase requesting the contributions of citizen linguists, again in a way that is compelling.

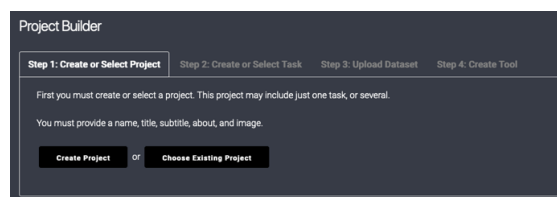


Figure 6: Project Builder

The project *description*, labeled “about your project and tasks” in the project builder, is typically a paragraph briefly describing the project research goals, how citizen linguists can help and what they will be asked to do. Where the previous fields could hold only plain text, this field accepts markdown, described in §5, to allow e.g. the use of links. Although a markdown capable field allows it, good design principles argue against complex formatting in the description given the space available. If the project has an external blog or web page, this can be entered in the News/Blog field and then reached via a News link.

Like the title, the project image should be representative of the project but also compelling to potential contributors. In addition, the project image should have an aspect ratio of 2 units high by 3 wide; that is, if the image were 200 pixels high it should be 300 pixels wide. Any multiple of 2x3 will display nicely however, images larger than 600 by 900 pixels will be scaled down (thus a waste of storage) while any smaller than ~ 200 x 300 will be scaled up and appear pixelated.

Project assets are media files uploaded not for annotation but to be included in e.g. the tutorial or reference guide.

Project designers can activate any of four discussion forums for their projects. The intended uses of the project forums are probably clear from their names. We anticipate that researchers will announce changes to the project, papers accepted, press coverage and other successes resulting from the use of project data in the Announcements forum. The *General Discussion* forum will most likely be populated by citizen linguists who discuss the project with each other. If the *Questions for Research Team* forum is activated then ideally the research team would monitor this on a regular basis and answer any questions arising from citizen linguists. Finally we have included a *Help and Technical Support* forum observing that in other citizen science portals, contributors often support each other which reduces the burden on the research team. Naturally, it would be wise to monitor this forum in case incorrect advice were given.

The *Research Team Members* section is a separate page, accessible from the project main page, that provides the names, titles, brief biosketches and images for the researchers who have developed the project. Similarly the Partner Badges section allows project designers to add the name, image and linked URL for each organizational partner. These appear at the bottom of the project main

page. Typically the image is a logo and links to the partner’s homepage.

With this information prepared, a researcher logs into LanguageARC using their account, which has previously been authorized as a project designer and clicks the Create a Project to access the Project Builder. The dialog box in Figure 6 will appear showing four tabs, the first labeled *Step 1: Create or Select Project* should be highlighted

Clicking “Create Project opens the New Project form in Figure 8. Only after completing this form, the project designer clicks Save.

Figure 8: New Project form

A few seconds later a dialog box should appear saying: *Project created or selected successfully*. Clicking the X dismisses the dialog box. Any information entered in Step 1 can be edited later, as described below.

5. Creating Tasks within a Project

If all has gone well so far, the project builder should have highlighted the tab *Step 2: Create or Select Task*. Clicking *Create Task* opens the *New Task* form shown in Figure 7.

Several fields on the *New Task* form will be familiar. A task requires an internal name, a title and call to action (labelled task description) that will appear on the project’s task list. Next, project designers can enter the contents of their *tutorial* and *reference guide*. Both of these open in new browser windows, giving the project designer more freedom in formatting. Both accept markdown that

can be used to insert formatting, links and media into the text following the specification linked from that form.¹ LanguageARC adds one new feature to the markdown specification: any file uploaded to the project assets can be inserted into any markdown capable field by surrounding it with *{local}* tags, e.g. *{local}MyAudio.wav{local}*.

Figure 7: New Task form

The next three fields require some explanation. With *Order of item assignment*, project designers can choose between assigning items in the order that they appeared in the manifest file or randomized uniquely for each contributor. If random is chosen, a second question will appear asking whether to allow repeats. Essentially, those are asking whether to performs the randomization with or without replacement. If *Repeating* is checked any single user may see some items multiple times before seeing all items in the data set.

The next question concerns whether to assign items within or across contributors. The former means that if a user were to see as many items as there are in the manifest they would actually have seen every item in the manifest. The latter means that the first batch of items will be given to the first contributor, and the next batch to the next contributor who requests them. In a task that had only one contributor, these would have the same effect. However if a second contributor joins the task before the first contributor has finished the first batch of annotations then the second contributor will receive the second batch. Various combinations of these choices allow a project designer to e.g. maximize the number of items that receive at least one imitation or to maximize the number of annotations an item receives.

The next two fields are familiar. A project designer may associate an image with the task that is different from the project image and from all other task images and may create a *General Discussion* forum specific to the task even if a *General Discussion* forum was created for the project as a whole. Only when the entire form is complete, the project designer clicks Save. If all goes well, a dialog box will appear saying; *Task created or selected successfully*. Clicking the small x will dismiss this dialog box. The Project Builder should highlight: *Step 3*

¹ <https://www.markdownguide.org/basic-syntax>

Upload Dataset. Any mistakes made in the Create Task form can be edited later as described below.

6. Creating a Dataset

As a reminder, a LanguageARC data set is a manifest file enumerating the items for some task with either item specific text or media files for each item. For projects that only require citizen linguists to answer questions or respond to simple prompts via speech, text or controlled vocabulary, the dataset could be composed of only a manifest containing those questions or prompts with IDs. For tasks that require contributors to listen to speech, read text or view images or video, the dataset would include all of the media segmented into files the right size for individual items as well as the manifest file that lists them all, assigns them IDs and optionally adds text specific to the items.

Although it is relatively simple to modify the fields in the project and task forms, LanguageARC does not allow a project designer to change a data set. There is a research reason behind this design decision. A significant change to a data set may render the contributions made after the change incompatible with the contributions made before. LanguageARC cannot predict when a dataset change is significant (and one might argue that researchers often cannot predict either). To underscore the importance of a dataset on research outcomes, LanguageARC assigns a unique number to each data set, even (especially!) datasets used for the same task, and records any change in dataset ID in the task's report. The only way to modify the data available to a task is to upload a new data set, even if only trivially different from datasets uploaded previously. Also, because LanguageARC allows multiple tasks to use the same data set, uploading a new data set does not erase an old one. In fact, LanguageARC does not currently include a function for erasing data sets given their importance to research outcomes. Obviously then care is required in the definition of a dataset not only because uploading multiple copies of the same data wastes storage on LanguageARC servers but also importantly because dataset changes in the midst of an ongoing task could impact research outcomes in ways that are hard to predict.

Selecting *Upload Dataset.* should open *New Dataset* form. The Dataset Name must be globally unique and should be memorable to the project team. The Dataset Description should describe dataset contents and use. For the next field, the project designer will click the Browse button, browse local, or any locally attached, storage to find the manifest file and upload it. The same process applies to uploading any media files except that the project designer should select and upload all files in a single pass. The final question offers a one-time randomization of the dataset order. Otherwise the dataset is order as specified in the manifest. This decision interacts with ordering and assignment decisions made when building the task. For example, a researcher who wants to provide the items in the same order to all contributors (for example for some surveys) would select no randomization of the dataset and when building the task would again select no randomization and assignment within contributors. If each contributor is to see a unique randomization of the items, it is sufficient to choose randomization when building the task. Only when the form is complete, clicking the Save button will create the dataset. The familiar dialog box

should appear saying: *Dataset created or selected successfully* and clicking the small x will dismiss it. If the dataset is very large in term of the number of size of files, creating the dataset may take longer than the previous steps.

7. Creating a Tool

To underscore the importance of tool design on research outcomes, LanguageARC assigns a unique number to each tool, records that change in a tool ID in the task's report and prohibits changes to a tool once created. The only way to change a tool is to first run a report to save all contributions made so far and then recreate the tool. As with dataset creation, care is required because any tool changes could impact the research outcomes in unpredictable ways.

With the project, task and dataset created, the project builder should have highlighted *Step 4: Create Tool.* The project designer should select Use Template to open the final Create Tool from Template form. There is a warning at the top that nothing is saved until the save button is clicked. Also, the project designer should not click Save until the form is complete. All fields on this form are new. The first asks for **exercise** specific text which can be thought of as instructions. They appear at the top of the tool and remain constant for all items in a task. A project can have multiple tasks each with different instructions but the instructions do not change within the task.

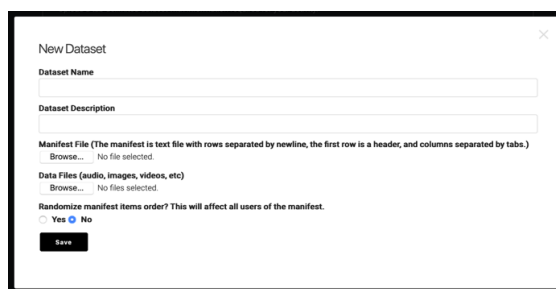


Figure 9: New Dataset form

The next field asks for the *Media Type*. The choice of text, audio, image, or video should match the type contained in the dataset. The 5th choice is labelled "manifest text" and indicates that there are no external media files and that all data for the task are included in the manifest. The third fields requires the project designer to select the column in which the media files are listed. Clicking on the arrow will pull down the list of the column headings in the manifest. If there are no media files any column can be selected.

Next, one decides whether the tool should offer a language selection. If the data and instructions make it clear that all tasks use a single language, then a language selector is not necessary. However, if the same activity can be done in multiple languages then 'yes' should be selected. A new field will appear indicating that there are two ways to add a language selector. The first is that the project designer can limit the range of languages to be selected by entering their names, each separated by a comma, in the text box. If the project designer chooses not to limit language selection, LanguageARC will load its universal language selector. This widget accepts all of the alternate names for all languages listed in the SIL Ethnologue. Each of these names indexes an official name and ISO code. The widget has look ahead so that as the user types the choices

decrease. Because the number of language names in the SIL Ethnologue is immense and because many languages have similar names, it is best to use this widget only when the true number of languages for a task is too large to enumerate.

The next field requires the project designer to select the manifest column containing the item IDs in the dataset. This is important as the IDs will appear in the automatically created report as the link between citizen linguists contributions and the dataset.

The next two form fields allow the project designer to indicate whether manifest columns contain item specific text to be displayed. Selecting yes causes two additional fields to appear, the first for the column in the manifest containing the item specific text and the second asking what label should appear above that text. LanguageARC accommodates two columns of item specific text, the primary appearing directly above the secondary.

The next fields allow the project designer to decide how the users will respond to each item. The first permits the response as audio. The corresponding widget includes record, stop and re-do buttons. Three additional fields offer a level test (currently deactivated), level meter and playback button. All audio is once the contributor clicks the record button followed by the stop button. The re-do button makes additional recordings. Researchers should attend to report that indicates whether the audio was re-recorded and act accordingly.

The next allows the project designer to accept responses as text. If selected, two additional fields appear asking how to label the response in the report and in the tool. When text response is activated a simple textbox appears in the tool under the label specified.

The next field, Judgement Buttons, allows the project designer to accept responses as controlled vocabulary. One enters text for each choice, one per line. If that field is empty, the tool will add a submit button so contributors can indicate when they have completed an item. If choices are entered, the Multiple Choice field becomes relevant. If no is selected, the judgments will appear as buttons and each will have the effect of a submit. In other words if the contributor clicks any button that decision will be saved and the tool will move to the next item. If instead yes is selected the decisions will appear as checkboxes, the contributor will be able to select one or more and a separate Submit button will appear which the contributor must click when they have finished making their decision. Project designers can include any or all of response audio, response text and judgement button but this feature should be used carefully. Including too many response modes may confuse contributors and make the data difficult to analyze.

The last two fields are radio buttons asking if the tools should allow skipping and reporting bad items. Selecting yes to the first will cause a red skip button to appear in the tool that contributors can click if they do not know how, or prefer not to, respond to the item. Selecting yes for the second will cause a red button labeled Report to appear inside the tool that contributors can click to indicate that there is something wrong with the item for example the audio is missing. Only when the entire form is complete should the project designer click Save. If all has gone well a small dialog box should appear saying that the tool has been created. Clicking the small X will dismiss this dialog.

8. Reviewing and Editing Projects

Clicking the Project link in the LanguageARC menu opens the project grid that should now include the newly created project, which will be visible only to the project team initially, probably on the last page of entries. On the project main page and task list, *Edit* links will appear only for authorized project designers (see Figure 10). Clicking the *Edit* link beneath the project menu on the left of the Project Main Page or Task List opens the *Edit Project Details* forms while clicking the *Edit* link beneath any task title will open the *Edit Task Details* form. All of the fields will be familiar from the New Project and New Task forms with two exceptions. The Position field allows the project designer to enter a integer to order projects in the grid or the tasks on the task list. The Project Status and Task Status pull downs allows the designer to change status from *Prototype* to *Private* and to *Request Publication*. A *Private* project or task is one intended to be permanently accessible by invitation only, to a controlled group of contributors.

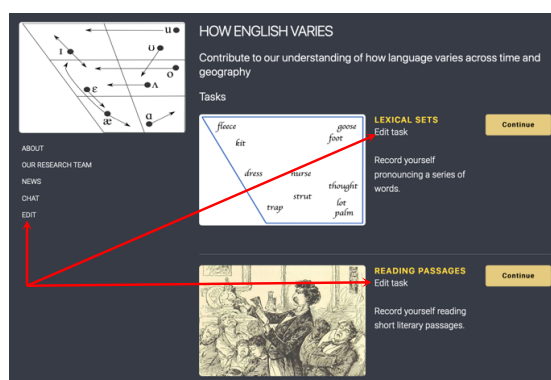


Figure 10: Links for Editing a Project or Task

To add Tasks to an existing project, an authorized project designer clicks the Create a Project link, but then selects *Choose an Existing Project* before selecting *Create New Task* and then continuing as described in §5 and following. It is possible to use an existing dataset in a new tasks if appropriate, for example to perform two different annotations over the same data in parallel. To do this the project designer would select *Choose Existing Dataset* rather than Upload Dataset at Step 3 in the Tool Builder. Although it is technically possible to upload a new data set for use with an existing task, given the interdependence of dataset and tool, this will require the task designer to Reset the Tool immediately after. This is not recommended for tasks in active use because of the possibility. Rather the task designer would be better served to prototype the new task and, when it is ready, invite users or request publication and then deactivate the old task by changing its status back to prototype. This will avoid confusing contributors and leaving the task in an undefined state and will keep the reports separate before and after the change.

9. Reporting

To report the results of a LanguageARC task, an authorized project designer clicks on their screen name in the upper right corner of any LanguageARC page. This opens the *Dashboard* as displayed in Figure 11. Clicking the Download Report button for the appropriate task will generate and download the report in TSV format in whatever way the browser is configured to accept it (e.g. save to a predefined folder, automatically open in a spreadsheet).

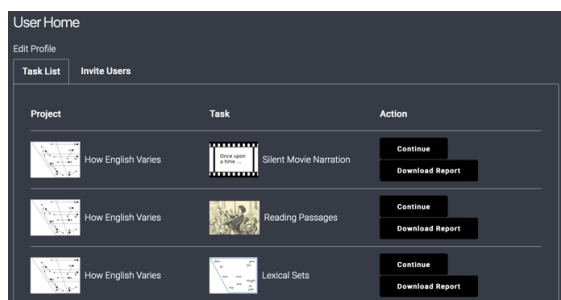


Figure 11: Dashboard

LanguageARC provides reports for every task using a consistent structure that begins with columns for the project ID and status, task ID and status, dataset ID, userID, country code and city from which the contribution was made followed by a date and time stamp using the GMT timezone. The remaining columns vary depending on the task. Figure 12 shows a tiny snippet of the report for a task to collect judgments of the home location of speakers based on their reading of an identical text, *Chicken Little*. The researcher who developed the project created multiple tasks to gather data on contributors' ability recognize the readers' social background and reports some of those results in this workshop (Cole 2020). Readers were from London, Surrey or Essex in the UK. Contributors could click a button to select one of those locations, skip the item, report it as bad (e.g. the audio was inaudible) or do nothing and simply exit the tool. The 11th and 12th columns contain the judgements contributed and the identifiers of the audio clips as the designer specified them in the manifest file. In the first row of the report snippet, the contributor exited the tool without making a judgment for clip 97. In the second, the contributor was offered audio clip 21 and clicked the Skip button. In the third row the contributor judged that the reader of clip 131 was from Essex.

Project ID	Project Status	Task ID	Task Status	Tool ID	Dataset ID	User ID	Country Code	City	Time	Judgment	Prompt ID
7	Published	24	Published	21	23	6	US	Fayetteville	2019-11-11 03:03:53 +0000		97
7	Published	24	Published	21	23	3	US	Philadelphia	2019-11-11 13:37:43 +0000	skipped	21
7	Published	24	Published	21	23	17	AU	Hobart	2019-12-03 12:41:48 +0000	Essex	131

Figure 12: A snippet of a LanguageARC report

One can also glean from the report that contributors come from diffuse locations, e.g. Philadelphia in the US and Hobart in Australia. This underscores the possibility that for a broadly available portal that tries to appeal to the public, there may be no time of day when a task is quiescent. It also shows that LanguageARC does not report locations any more specific than the city. This is to further protect the anonymity of contributors.

10. Conclusion

This paper has described to goal, features and operations of LanguageARC, a portal designed to allow researchers to easily create projects and tasks that attract citizen linguists who are motivated by their interest in language and in the individual projects and by the opportunity to join with like-minded people, to learn about and make small contributions to those projects. This approach augments existing approaches that rely principally on monetary incentives to motivate contributions. By coordinating efforts that use these complementary approaches we will be able increase the number, scale and diversity of language resources in order to promote language related education, research and technology development.

11. Acknowledgements

LanguageARC is an outcome of the NIEUW project to investigate novel incentives and workflows in the elicitation of language data. Other NIEUW outcomes include the LingoBoingo.org language games portal and the language identification game, NameThatLanguage.org. The Linguistic Data Consortium and the University of Pennsylvania acknowledge the generous support of the US National Science Foundation via the Computer and Information Science and Engineering Directorate's Research Infrastructure program, grant 1730377.

12. Bibliographical References

- Cieri, Christopher, Mark Liberman, Stephanie Strassel, Denise DiPersio, Jonathan Wright, Andrea Mazzucchi, James Fiumara (2018) From 'Solved Problems' to New Challenges: A Report on LDC Activities. In Calzolari, et. al., Proc. 11th International Conference on Language Resources and Evaluation (LREC 2018), pp. 3265-3269.
- Christopher Cieri, James Fiumara, Mark Liberman, Chris Callison-Burch, Jonathan Wright (2018) Introducing NIEUW: Novel Incentives and Workflows for Eliciting Linguistic Data. In Calzolari, et. al., Proc. 11th International Conference on Language Resources and Evaluation (LREC 2018), pp. 151-155.
- Cole, Amanda (2020) Identifications of Speaker Ethnicity in South-East England: Multicultural London English as a Divisible Perceptual Variety In Proceedings of the Citizen Linguistics for Language Resource Development workshop at LREC 2020.