2025-03-31

Consistent Translations Glossary across R

Saranjeet Kaur Bhogal
Imperial College London

Signatories

Project team

Saranjeet Kaur Bhogal will be the lead developer for this project. She is a Research Software Engineer in the central Research Software Engineering team at Imperial College London and has experience in developing R packages and contributing to the R community. She will be responsible for developing the proposed R package and coordinating with stakeholders, in particular the R Contribution Working Group (RCWG), to ensure the package integrates with their translation process. Saranjeet is a member of the RCWG, where she has co-authored the R Development Guide and has also contributed to translations of R messages to Hindi. She has also contributed to the R community through the R Dev Days.

Contributors

This proposal has benefitted by the contributions of the following people (in alphabetical order):

- Diego Alonso Alvarez
- Gergely Daróczi
- Heather Turner
- Maëlle Salmon
- Michael Chirico
- Yanina Bellini Saibene

Consulted

The following people (in alphabetical order) have been consulted and have provided feedback on the proposal:

- Diego Alonso Alvarez

- Gergely Daróczi
- Heather Turner
- Maëlle Salmon
- Michael Chirico
- Yanina Bellini Saibene

The Problem

There are many volunteer translation efforts across the R community (messages in base R/recommended packages, individual packages, rOpenSci materials such as their package development guide, Bioconductor training materials, R books such as R for Data Science, etc). These projects often have their own glossaries, since the benefits of using a glossary in translation projects are widely recognised (improves translation quality, reduces translation effort). However, developing and maintaining separate glossaries for each translation project reduces these benefits - different projects may use different translations for certain terms, effort to identify relevant terms and translate them is duplicated across projects putting unnecessary burden on volunteers. Some projects do not have the volunteer capacity to translate the glossary into multiple languages, or may not even use a glossary. Moreover, currently it is difficult for projects to take full advantage of existing glossaries due to diversity in the choice of translation infrastructure. These issues will become even more important if internationalization of R help pages becomes more common (ref R Consortium-funded project: tooling-for-internationalization-of-r-help-pages)

An example of diverging translation of glossary terms is found in the rOpensci English-Spanish glossary:

English	Spanish (rOpenSci)	Spanish (R project)
path	ruta de acceso	ruta (file path) / trayecto (polygon path)

It's possible that's due to the term arising in different contexts; nevertheless, there are many examples of diverging translations that might have been avoided if a glossary had been used. For example, in base R, "loop" is variously translated as "bucle", "loop", "ciclo", though both the rOpensci and R project glossaries agree on "bucle" being the preferred term; sometimes "camino" is used rather than "ruta" for a file path. These don't change the meaning but may read a bit strangely.

This proposal aims to address this issue by creating a data package in R package that will help maintain a consistent glossary to improve the quality of translations. This will help in making R more accessible to users who do not speak English and improve the overall user experience of R. Previously this problem has been discussed within the R Contribution Working Group (RCWG)

The proposal

The R Contribution Working Group (RCWG) has been involved in a number of projects to foster a wider, more diverse, community of contributors to base R. The group facilitates active contribution by the community, especially through the R Dev Days. This proposal is specifically related to efforts to support contributions to the translations of messages in R. The aim is to improve the technical

infrastructure of the translations process by providing a common glossary that can be used by all translation projects in the R ecosystem, thus avoiding duplicated work and effort.

Overview

The goal of this project is to improve the technical infrastructure of the translations process by providing a common, shared glossary across different translation projects in the R ecosystem. It will help in improving the quality of translations and make it easier for users to contribute to the translations process. This will help in making R more accessible to users who do not speak English and improve the overall user experience of R.

At present, different R projects use different translation workflows:

- base R and the recommended packages - use Weblate, using a common glossary across base R and recommended packages.
- rOpenSci - uses a GitHub-based workflow. They also have a work in progress glossary.
- Bioconductor - uses Crowdin and is most likely not using any glossaries.

One-off projects, such as translating an R book or translating messages in individual package, could use one of the above or yet another workflow, as the translators prefer.

Hence, we want to store the glossary and translations in an R package, provide a process for proposing changes outside of Weblate, and provide tools for syncing with Weblate (possibly also Crowdin), potentially benefiting base R and recommended packages, rOpenSci, Bioconductor, and others working on internationalization within the R community.

Detail

The R Contributors, as a part of R Dev Days, have been compiling a combined list of terms from the language-specific glossaries on the R Project Weblate server and flagged 175 terms as "terminology" in the English glossary. This flag ensures this common set of terms appear in all the language-specific glossaries. Most languages are using essentially these terms (most have a glossary of 178 terms, including a few that are still in the glossary without the "terminology" flag). The Translation Team leads can also add further terms to the terminology set.

This project aims for the following:

- Creation of common glossary, starting by combining Weblate glossary terminology terms and rOpenSci WIP glossary:
  - Columns to include: date added, source string, translations for different languages, optional explanation for different languages (e.g. can explain that a particular word should be kept in English, or note synonyms), explanation in English (can be used to explain the meaning of a term), added via (Weblate/GitHub), updated via (Weblate/GitHub), Weblate (include/exclude/NA).
- Documented process for people to suggest additions to and deletions from the common glossary:
  - Preferred: pull request on the underlying CSV so that it is easy to update the common glossary once proposed changes are reviewed. A benefit of having the CSV format is the glossary can be re-used by people independently of the proposed package. Another approach would be to use a YAML format that can be compiled into a CSV, if needed. When using a

YAML with support for lists, each translation could have its own line, and it could include a lot of metadata (if needed) as it would not be restricted by the tabular format,
  - Alternative: proposal via GitHub issue, perhaps using template to ensure essential information is given.
- Tools to update common glossary with updates from Weblate:
  - Get current glossary and translations from Weblate,
  - Update common glossary with new terminology and their translations (set `added_via` column to Weblate and `weblate` to include),
  - Update common glossary with new/updated translations of existing terms (set `update_via` column to Weblate),
  - Update common glossary with deleted terms. These terms should not be removed, but the `weblate` column should be set to exclude),
  - GitHub Action to regularly update common glossary based on updates to Weblate.
- Tools to update Weblate with changes in common glossary:
  - Get current glossary and translations from Weblate,
  - Update Weblate glossary with terms added via GitHub that are flagged to be included (should be added to English glossary with flag "terminology"),
  - Update Weblate glossary with new/updated translations of terminology terms,
  - GitHub action to regularly update Weblate based on changes to common glossary. Need to ask Gergely if this is possible - may need to be semi-manual process.
- Tools for CrowdIn:
  - Could be similar to those for Weblate, but currently no glossary to test on. Could perhaps start one on an active project.
- Maintenance tools (for tasks requiring manual review, that may need doing from time to time):
  - Identify strings in common glossary that have been removed from Weblate (`added_via` is Weblate but `weblate` is `exclude`). Select to keep (change `added_via` to GtiHub) or delete (remove term from common glossary),
  - Identify strings in common glossary that are not on Weblate (`added_via` is GitHub and `weblate` is NA). Select to add (set `weblate` to include) or ignore (set `weblate` to `exclude`),
  - Identify strings in the Weblate glossary that are not flagged as "terminology". Select to add terminology flag or remove,
  - Identify strings in the Weblate glossary that are not flagged as "terminology". Remove from the Weblate glossary for all languages,
  - Identify strings in that are in language-specific glossaries but not in the English glossary. Select to add terminology flag or remove (language leads can propose new terms, but should end up using for all languages or none).

Design principles: - Automate synchronously as much as possible, but allow control over what is included in each glossary.

Minimum Viable Product

A data package on R-universe that contains the common glossary and translations. The package will have a function that will allow users to update the glossary. The package will also have a function that will allow users to sync the glossary with Weblate.

Architecture

The package will use the Weblate API. The package will have a data frame that will store the glossary. The package will also have a function that will allow users to update the glossary.

Assumptions

The Weblate API is made available and accessible to the developer. This has been discussed in the relevant issue.

Sustainability

This project will be maintained by the R Contribution Working Group. A process will be provided for people to suggest additions to and deletions from the common glossary.

Project plan

Start-up phase

The project will be setup as a GitHub repository and the various steps for the development will be opened as issues on GitHub. Development will take place and be tracked by creating pull requests associated to the issues. Appropriate license will be chosen for the package. The contributors to the project will be acknowledged using the all contributors bot on GitHub. The work on the project will be regularly reported to the community through the R Contribution Working Group meetings and occasionally through the R Contributor Office Hours.

Technical delivery

The work on the package development is scheduled for July - November 2025.

Timeline	Tasks and milestones
First half of July 2025	Setup the technical infrastructure and create issues on the GitHub repo
Mid July 2025	Present the initial setup to the RCWG and get feedback
Second half of July 2025 - September 2025	Development work
End of September 2025	Mid project presentation to the RCWG and get feedback
October 2025 - November 2025	Continue the development work based on the feedback
End of November 2025	Final presentation to the RCWG, write and share a blog post about the work

The development work will be associated with milestones and deliverables which will be tracked through GitHub. A final blog post will be published to share the work done along with social media announcements. If possible, the work will also be publicised at relevant events, including R-Ladies+ and RUG meetups.

Other aspects

The project will be promoted through the RCWG to ensure that the R community is aware of the project. Regular updates will be provided to the R community on the progress of the project.

Feedback will be sought from the community to ensure that the package meets the needs of the users. The updates will be provided through regular blog posts and announcements on social media platforms. The project will also be discussed with local R-Ladies+ and R User Groups.

Requirements

1. **People**: The lead developer will be responsible for developing the R package and coordinating with the RCWG.

2. **Processes**: The project will follow the best practices for R package development and will seek feedback from the R community to ensure that the package meets the needs of the users.

3. **Tools & Tech**: The project will use several packages in R that support the package development process (devtools, testthat, roxygen2, pkgdown), GitHub, and the Weblate API.

4. **Funding**: The project requires funding to support the development of the R package. The funding will be used to cover the costs associated with the development, testing, documentation, and release of the package by the lead developer.

The project will require coordination between the lead developer, the RCWG, and the R community to ensure that the package meets the needs of the users and is easy to use.

People

Saranjeet Kaur Bhogal, who is a Research Software Engineer at Imperial College London, will be the lead developer for this project. She has experience in developing R packages and has contributed to the R community. She will be responsible for developing the R package and coordinating with the RCWG to ensure that the glossary is consistent across languages. Saranjeet is a member of the RCWG, where she has co-authored the R Development Guide and has also contributed to translations of R message to Hindi.

The RCWG will be involved in this project to provide guidance and support. The RCWG is a group of volunteers who are actively involved in improving the R ecosystem and have experience in translation projects.

Feedback on the work will be sought from the R community at large, especially from users who are involved in the translations process. This will help in ensuring that the package meets the needs of the community and is easy to use.

Processes

The project will follow the following processes:

1. **Development**: The R package will be developed using the R package development tools (`devtools`, `testthat`, `roxygen2`). It will be hosted on GitHub and will follow the best practices for R package development.

2. **Translation**: Feedback will be sought from the R community to ensure that the glossary is consistent across languages.

3. **Community engagement**: Regular updates will be provided to the R community on the progress of the project through the RCWG meetings and occasionally through the R Contributor Office Hours. Feedback will be sought from the community to ensure that the package meets the needs of the users.

4. **Handover and maintenance**: Eventually, the will be maintained by the RCWG.

5. **Code of conduct**: A code of conduct will be put in place to ensure that the project is a safe and welcoming space for all contributors.

Tools & Tech

The project will use several packages in R that support the package development process (devtools, testthat, roxygen2, pkgdown), GitHub, and the Weblate API. Most of these tools are open-source and widely used in the R community. The Weblate API is used by the RCWG to track and translate messages in R.

Funding

The project seeks funding from the R Consortium Infrastructure Steering Committee (ISC) to support the development of the R package. The funding will be used to cover the time that the lead developer will invest in the development of the package.

The total funding requested for the project is $8,000.

Summary

The project requires funding to support the development of a data package in R and streamlining the translation process for the R community. The lead developer will be responsible for developing the R package and coordinating with the RCWG to ensure that the glossary is consistent across languages. The project will follow best practices for R package development and will seek feedback from the R community to ensure that the package meets the needs of the users. Thus improving the translations infrastructure in R and making it easier for users to contribute to the translations process. The funding will be used to cover the time that the lead developer will invest in the development of the package.

Success

The project will be considered a success if the following criteria are met:

- The key deliverables are met: the package includes the planned functionality, which is documented and tested.
- The documentation is available on the package website.
- At least 75% of the open issues on the planned work are closed or at least become work-in-progress.
- Feedback is sought (and incorporated as far as possible) from the R community.
- At least two community members that are not directly involved in the project contribute to the project, by submitting issues, making a pull request or making a commit to the git repository.

Definition of done

The project will be considered done when the following criteria are met:

- The R package is developed and tested.
- The package is documented and available on the package website.
- Feedback will be sought from the R community to ensure that the package meets the needs of the users.

Measuring success

The success of the project will be measured by the following metrics:

- At least 75% of the open issues on the planned work are closed or at least become work-in-progress.
- At least 85% of the pull requests are reviewed and merged.
- The package is documented and available on the package website.

Future work

The project can be extended in the following ways:

It would be beneficial to refer to what has been previously built. For example, as a source, rOpenSci consults glosario, a community-curated glossary of data science terms, although it has a broader scope as a glossary. Hence, it would be worth considering syncing with glosario to ensure that the R glossary is consistent with the broader data science community. Thus, this work would benefit not only the R community but also the broader ecosystem. Given the broader scope of glosario, a two-way sync would be beneficial. For instance, a two-way sync for terms in the R glossary that are also in glosario, i.e.

copy over translations and explanations from glosario to the R glossary (under CC-By-4.0 license, attribution could be given in the documentation for the new package), where these are missing in the R glossary contribute any translations back from the R glossary to glosario Applying a CC0 licence to the glossary would support in freely sharing the glossary with other projects.

Notably, there is an R package associated with glosario (glosario-r), which is a WIP but can be used to retrieve glossaries.

Key risks

The are not any major risks associated with this project. The idea for the project is based on an actual need by the R community and has the support of the RCWG. The lead developer has experience in developing R packages and has contributed to the R community. The project will follow the best practices for R package development and will seek feedback from the R community at appropriate stages.