

Contributing to Glosario

Last updated: Aug 19

Video: Quick [guide in English on how to contribute to Glosario](#).

Video: Quick guide to [Glosario contribution in Spanish](#).

[Glosario](#) is The Carpentries' open-source glossary of terms used in **data science**. By adding glossary keys to a lesson's **metadata**, authors can indicate what the lesson teaches, what learners should know before they start, and where they can find it. Authors can also use the **library's functions** to insert consistent hyperlinks for terms and definitions in their lessons in several languages. This contribution guide provides Glosario contributors with step-by-step instructions for adding definitions of data science terms to the glossary.

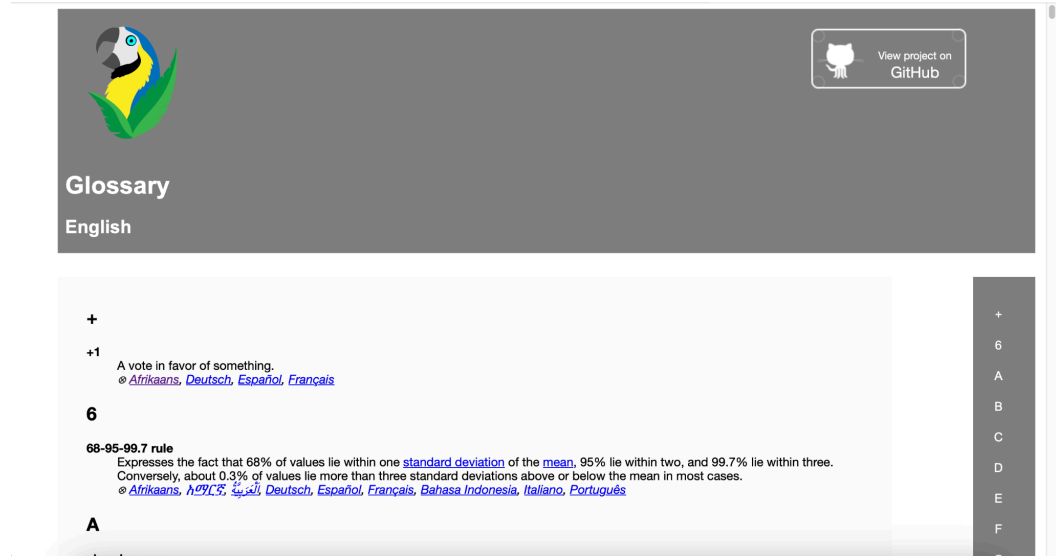
Adding a definition in another language to an existing term

Before you start:

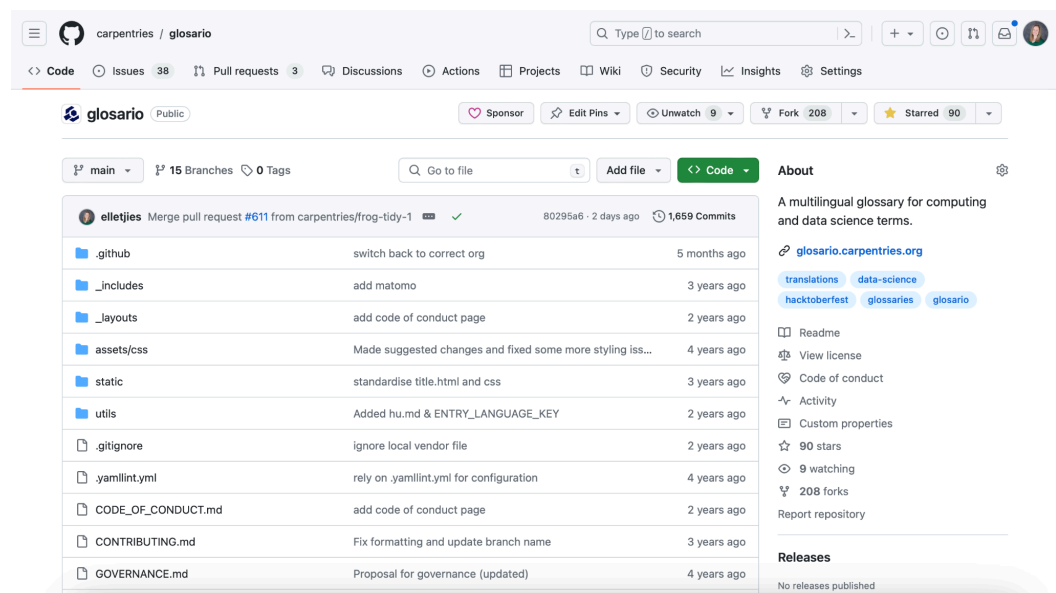
- Check the [open Issues](#) and [Pull Requests](#) to see what terms and definitions other contributors are already working on. Doing so will help you avoid accidentally spending time writing the same term(s)/definition(s) as someone else.

To add a new entry:

1. Go to the [Glosario website](#) and decide which term you want to translate. For convenience, you can view the English glossary to see which terms have already been translated. The image below shows that the term “+1” has been translated into Afrikaans, German, Spanish and French. The term “68-95-99.7 rule” has been translated into Afrikaans, Amharic, Arabic, German, Spanish, French, Bahasa Indonesia, Italian and Portuguese.



- a.
2. Login to [Github.com](https://github.com) and open the [main Glosario repository](#).



- a. If you have already **forked** the Glosario **repository** into your account, then please skip ahead to step 3.
- b. If you have not **forked** the Glosario **repository**, click the **Fork** button near the top right of the page (not the arrow icon right next to it!). At the next screen, click Create **Fork** without changing any options:

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk ().*

Owner * Repository name *

 froggleston / glosario

✔ glosario is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

A multilingual glossary for computing and data science terms.

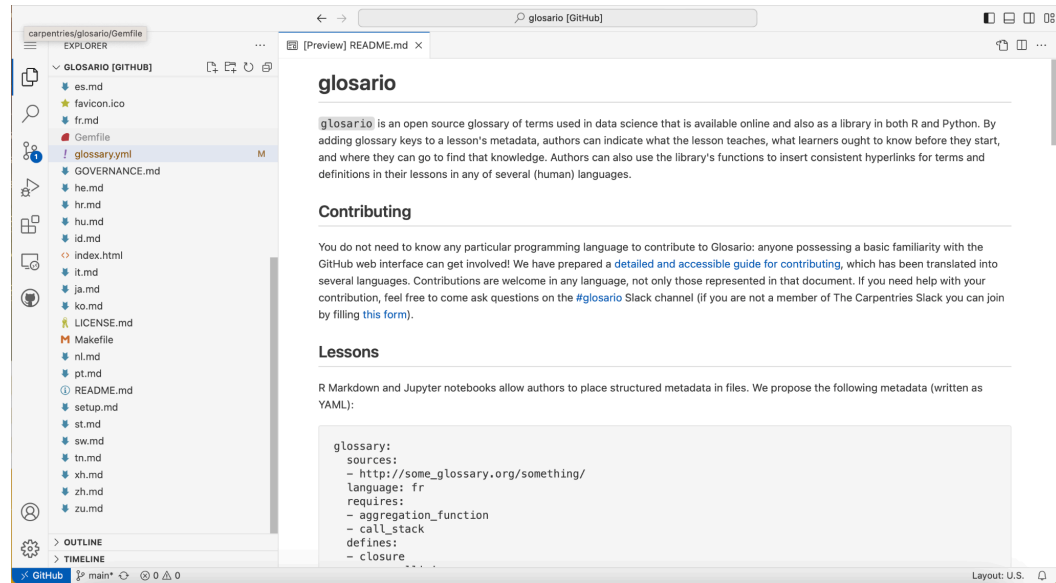
☒ Copy the `main` branch only

Contribute back to carpentries/glosario by adding your own branch. [Learn more.](#)

 You are creating a fork in your personal account.

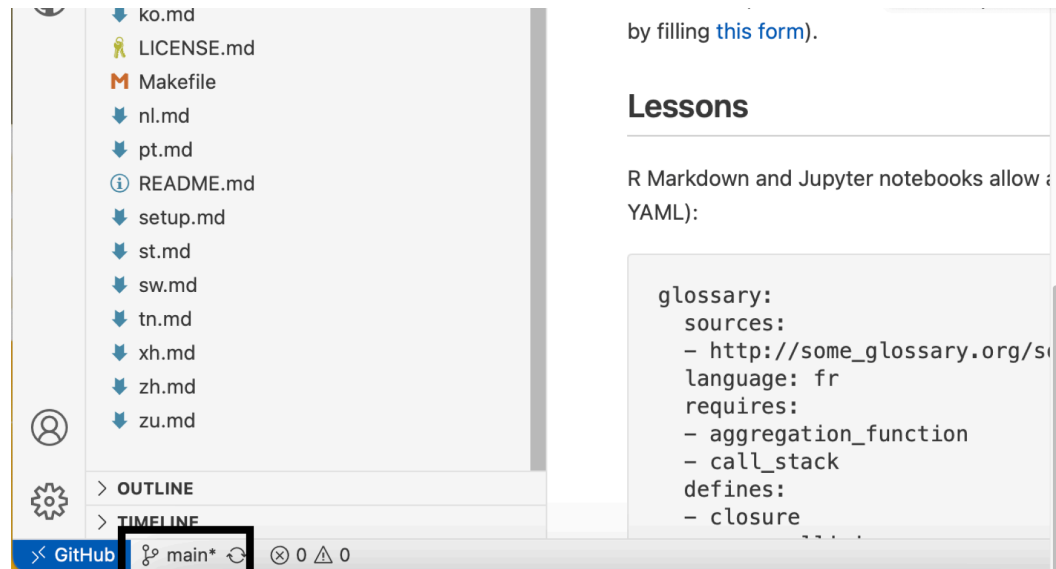
Create fork

- c.
- d. Once your **fork** has been created and Github takes you to your **forked** version of the Glosario **repository**, proceed with step 3.
3. Hit the full-stop/period key on your keyboard. GitHub will automatically create a new **codespace** environment and open the web-based [vscode](#) code editor. A page like the one below should appear after a time:
 - a. This works like a desktop tool, with a folder browser to the left and the main code editor window to the right.
 - b. The main menu can be accessed by clicking the three-line **hamburger icon** in the top left.
 - c. Right-clicking in certain areas will bring **contextual menus** like your operating system desktop.



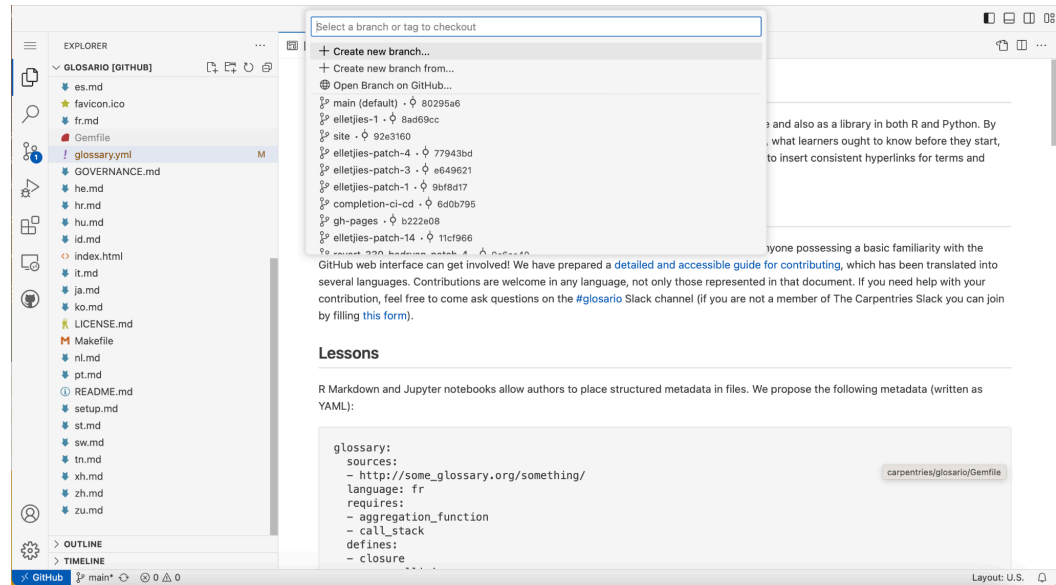
d.

4. Click the “main” button at the bottom left of the page to create a new **branch** for your Glosario contribution.

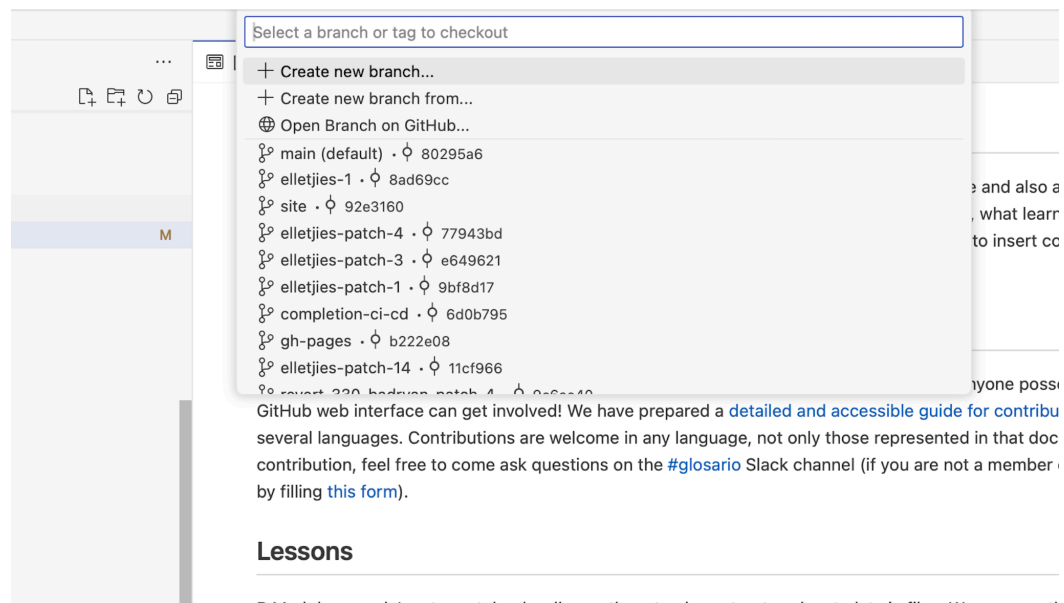


a.

- b. A box will appear at the top of the page where you must click “Create new branch”.



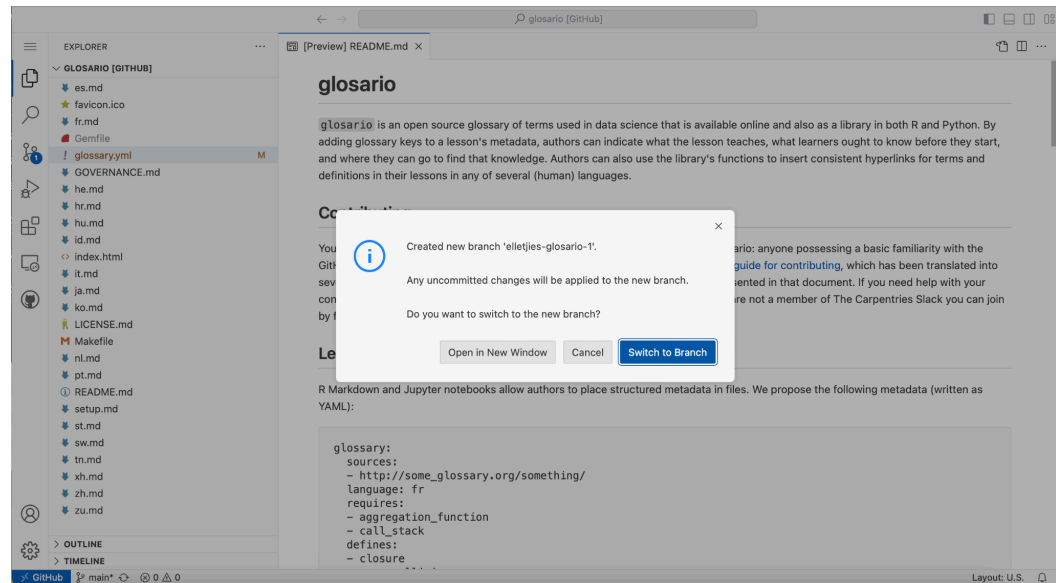
C.



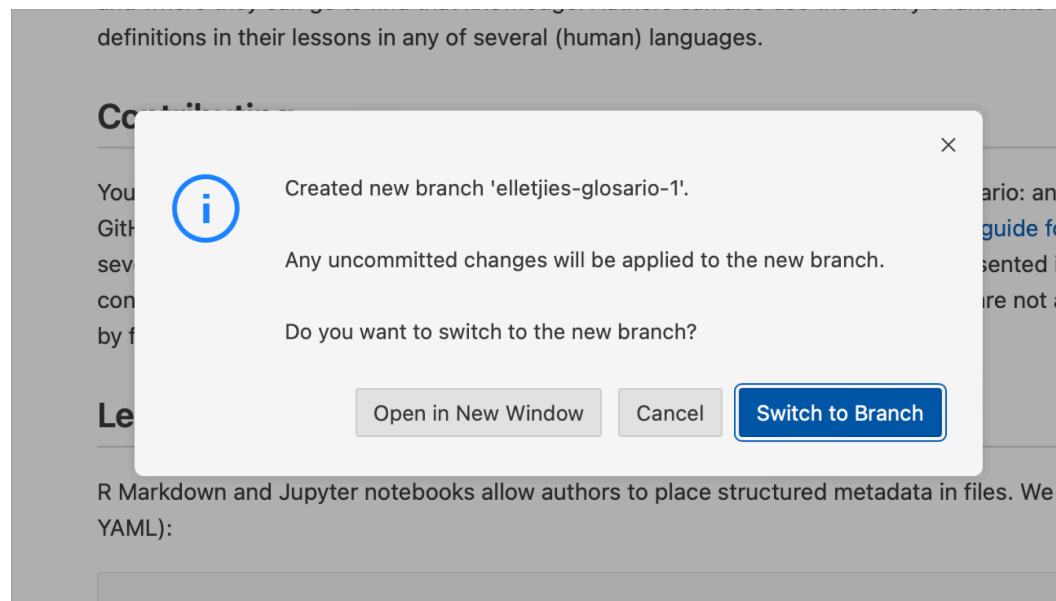
d.

5. Enter a name for your **branch**:

- Branches** should be in kebab case, usually containing the username of the person who maintains the **branch** (you! in this instance *elletjies*), and the name should be concise but specific. For example, when adding a translation to Glosario, one might call the **branch** *elletjies-glosario-1*.
- Once you have added a name, hit **Enter** and click **Switch to Branch** in the following popup.

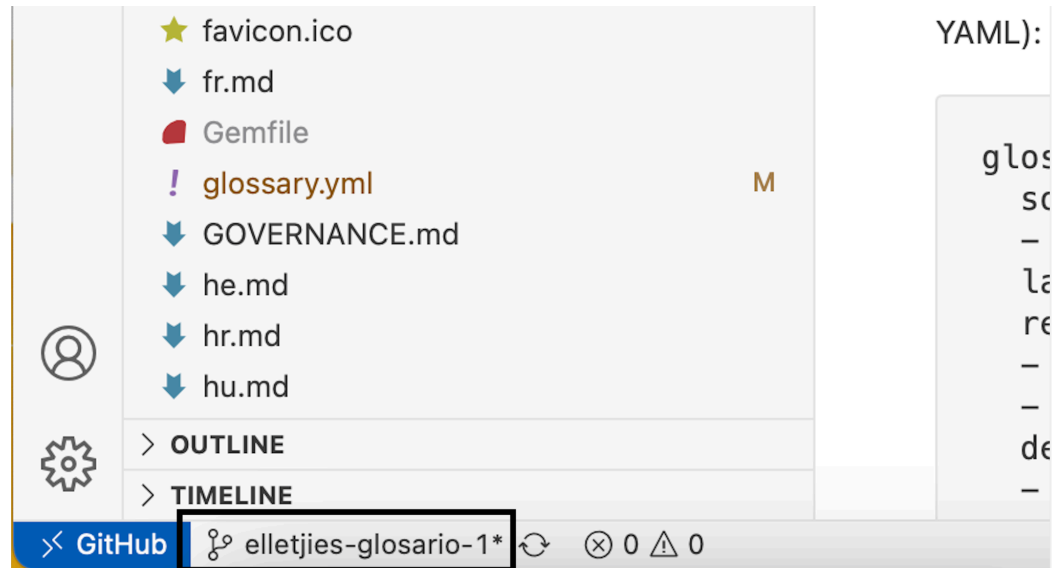


C.



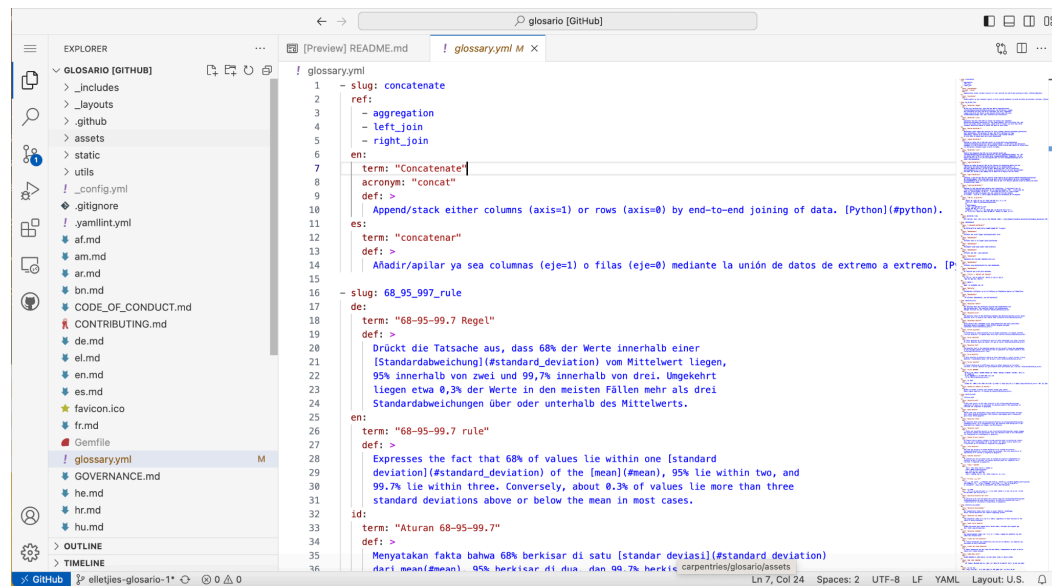
d.

6. Once the **branch** creation is complete, the main button in the bottom left should now contain the name of your new **branch** (*elletjies-glosario-1*).

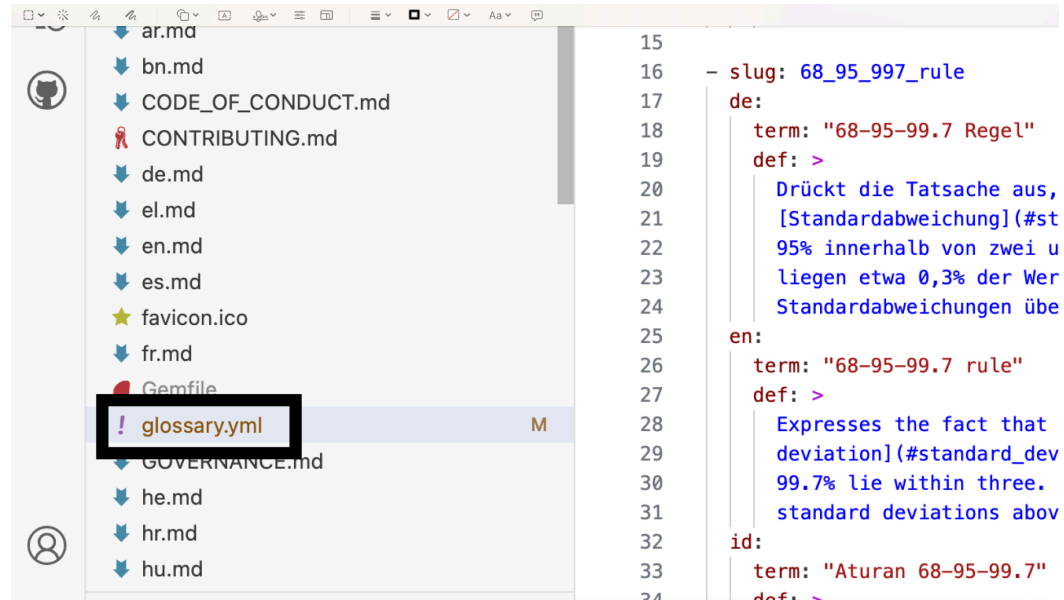


a.

7. Now, use the folder browser on the left to access the *glossary.yml* file.



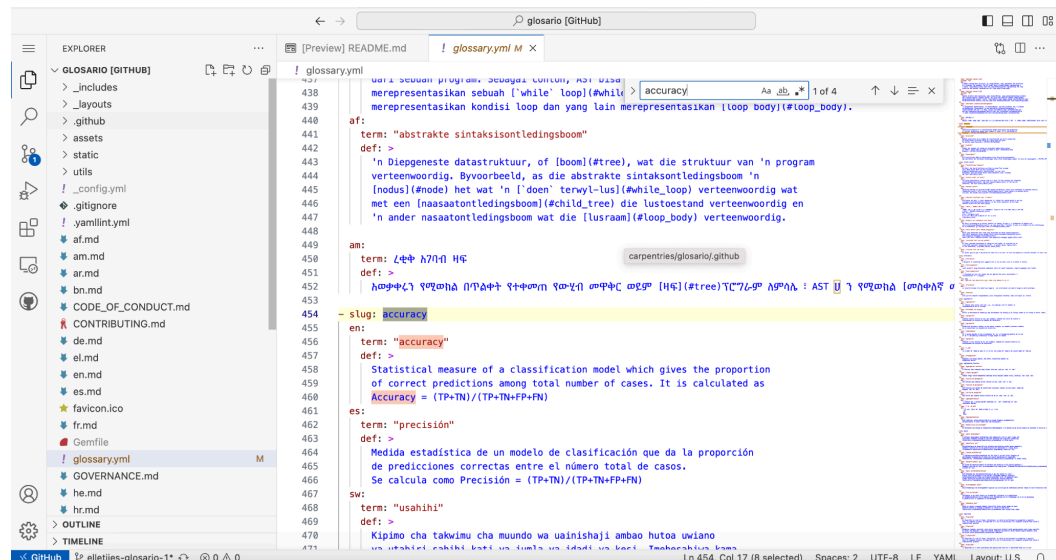
a.



b.

8. Adding your translation:

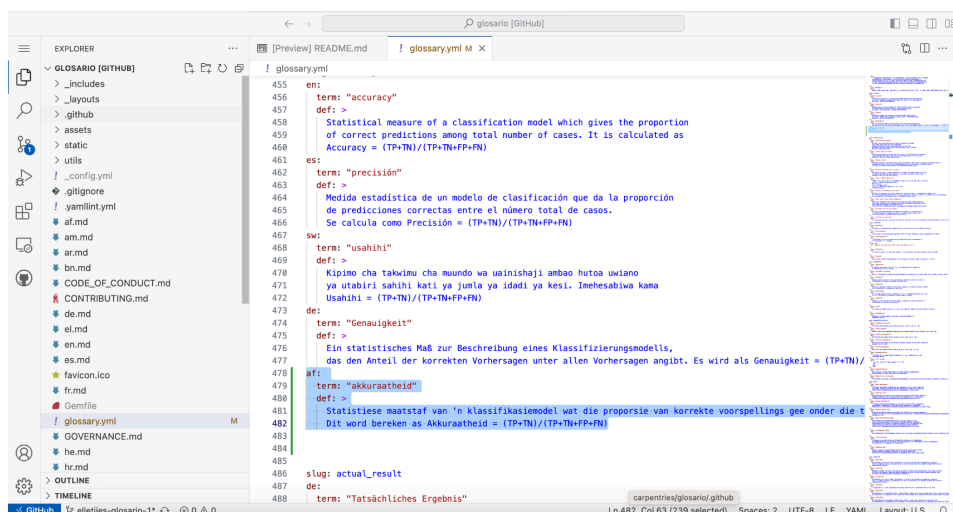
- Use the *Ctrl + F* function to find the term you want to translate. If a term is found already in the glossary that you want to translate, the line will be highlighted.
- In this example, we would like to translate the English term “accuracy” into Afrikaans (af)



c.

- Scroll down to the last translated term, in this case, *German (de)*

- e. Each translation has a unique **slug**, and several translations are comprised of a [language code](#), a term, and a definition. Note that the *glossary.yml* file format (YAML) needs a specific indentation to be used correctly, as follows:
- i. The first indentation level (no spaces, aligned to the left-hand-side) is the **slug** label, a colon, and the **slug** for the translations, e.g. `slug: accuracy`, and each translation's language code followed by a colon, e.g. `en:`
 - ii. The second indentation level (two spaces in, aligned to the first vertical line) is the term label, a colon, and the term in English in double quotes, e.g. `term: "accuracy"`, and on the following line the definition label, a colon, and a greater than an arrow, e.g. `def: >`
 - iii. The third indentation level (four spaces in, aligned to the second vertical line) is the definition where your translation should be added.
- f. You can now start to add your translation to the **glossary.yml** file. Supply your [language code](#), term, and def terms as seen in the glossary.yml file.
- i. The language **ISO code** needs to be added; in the case of Afrikaans, it should be `"af:"`; in this example, it appears **red**, then press Enter.
 - ii. Add two spaces and add `"term"`; the term will remain **red**, the `":"` will turn **blue** (in this example), and the translated term `"akkuraatheid"` (in double quotes) should be **red**, then press Enter.
 - iii. Add `"def: >"`. `"def:"` and `">"` will remain **red**, but the `":"` will be **blue**, then press Enter.
 - iv. Add four spaces. Now, you can add your translation. The words should appear in **blue**.



V.

- vi. Ensure there is one blank line between the end of each block of translations in a **slug**. In the example below, there are too many lines.

```
af:
  term: "akkuraatheid"
  def: >
    Statistiese maatstaf van 'n klassifikasie-model wat die
    Dit word bereken as Akkuraatheid = (TP+TN)/(TP+TN+FP+FN)

slug: actual_result
de:
  term: "Tatsächliches Ergebnis"
```

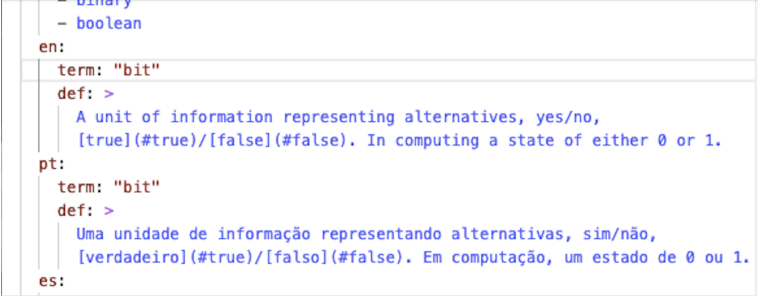
1.

- vii. **NOTE:** If your translation includes terms with **hyperlinks**, see the example below:

```
bit
  A unit of information representing alternatives, yes/no, true/false. In computing a state of either 0 or 1.
  → binary, Boolean
  ⌘ Afrikaans, Akan, Deutsch, Español, Français, Bahasa Indonesia, Português, Nederlands, Kiswahili, Setswana
```

1.

2. In the *glossary.yml* file, it looks like this:

- a. The screenshot shows a code editor with a light blue background. It contains a translation entry for the term 'bit'. The entry is structured as follows:
- A list of languages: 'binary' and 'boolean'.
- A section labeled 'en:' containing:
 - 'term: "bit"'
 - 'def: >'
 'A unit of information representing alternatives, yes/no, [true](#true)/[false](#false). In computing a state of either 0 or 1.'
- A section labeled 'pt:' containing:
 - 'term: "bit"'
 - 'def: >'
 'Uma unidade de informação representando alternativas, sim/não, [verdadeiro](#true)/[falso](#false). Em computação, um estado de 0 ou 1.'
- A section labeled 'es:' at the bottom, which is currently empty.

3. For your translation, follow this convention:

- a. **Translate** the term in square brackets and **leave** the text in round brackets in English. See the Portuguese (pt) example.

9. Once you have reviewed your translation and checked the formatting against the other translations (make sure your entry matches the vertical lines connecting the language code, `term`, and `def`, which help you get the right indentation), click the Source Control icon in the button menu on the far left of the screen (it should be underneath the magnifying glass icon and should have a circle with a number in it denoting the number of changes to **commit**).

- a. Enter a message for the **commit** in the box in the top left, e.g.,
‘Translating “accuracy” into Afrikaans.’
- b. Click **Commit** and **Push**.

c.

```

455 en:
456   term: "accuracy"
457   def: >
458     Statistical measure of a classification model which gives the proportion
459     of correct predictions among total number of cases. It is calculated as
460     Accuracy = (TP+TN)/(TP+TN+FP+FN)
461 es:
462   term: "precisión"
463   def: >
464     Medida estadística de un modelo de clasificación que da la proporción
465     de predicciones correctas entre el número total de casos.
466     Se calcula como Precisión = (TP+TN)/(TP+TN+FP+FN)
467 sw:
468   term: "usahihi"
469   def: >
470     Kipimo cha takwimu cha muundo wa uainishaji ambao hutoa uwiano
471     ya utabiri sahihi kati ya jumla ya idadi ya kesi. Inehesabiwa kama
472     Usahihi = (TP+TN)/(TP+TN+FP+FN)
473 de:
474   term: "Genauigkeit"
475   def: >
476     Ein statistisches Maß zur Beschreibung eines Klassifizierungsmodells,
477     das den Anteil der korrekten Vorhersagen unter allen Vorhersagen angibt. Es wird als Genauigkeit = (TP+TN)/
478 af:
479   term: "akkuurtheid"
480   def: >
481     Statistiese maatstaf van 'n klassifikasie-model wat die proporsie van korrekte voorspellings gee onder die t
482     Dit word bereken as Akkuurtheid = (TP+TN)/(TP+TN+FP+FN)
483
484 slug: actual_result
485
486 de:
487   term: "Tatsächliches Ergebnis"
  
```

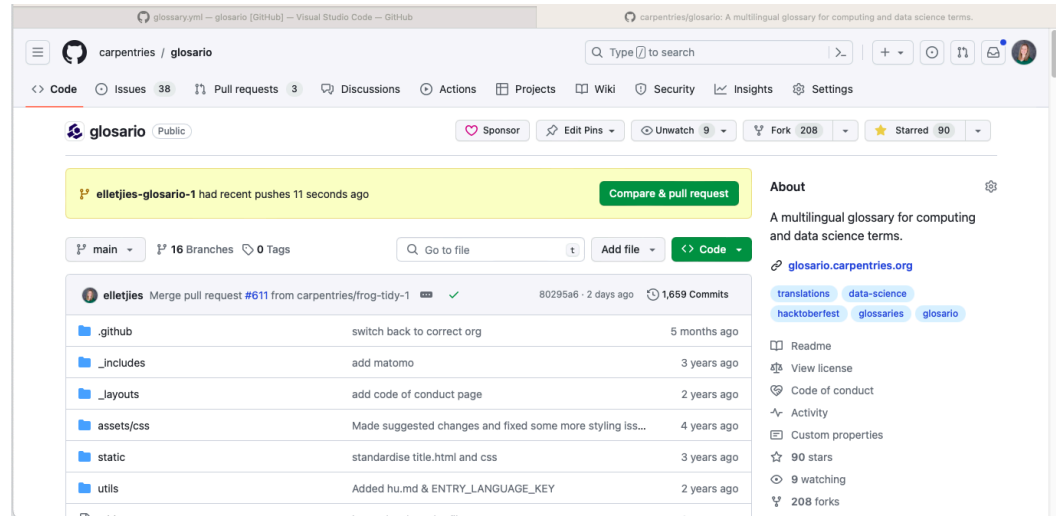
d.

```

455 en:
456   term: "accuracy"
457   def: >
458     Statistical measure of a c
459     of correct predictions amo
460     Accuracy = (TP+TN)/(TP+TN+
461 es:
462   term: "precisión"
463   def: >
464     Medida estadística de un m
465     de predicciones correctas
466     Se calcula como Precisión
467 sw:
468   term: "usahihi"
  
```

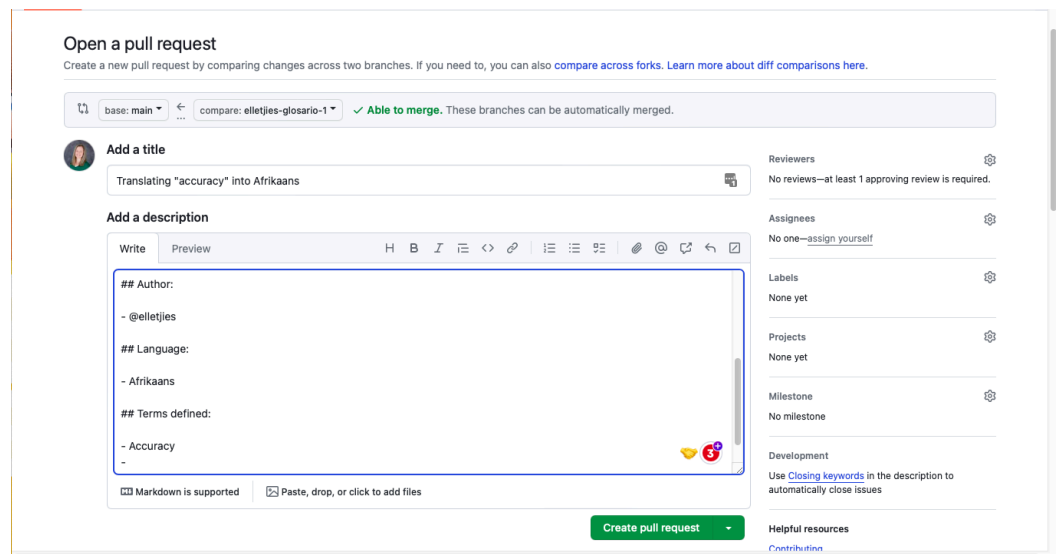
10. You now need to exit the **codespace** environment. Access the main menu by clicking the three-line **hamburger** icon in the top left.

- Click on “Go to Repository.”
- You will now see your **pull request**; click “Compare & pull request.”



c.

- d. Add a title; in this case, we used ‘Translating “accuracy” into Afrikaans’ and added a description. Add your GitHub handle, the language you have translated into, and the name(s) of the defined term(s). You can now click on “*Create pull request.*”



e.

11. Congratulations - you are done!

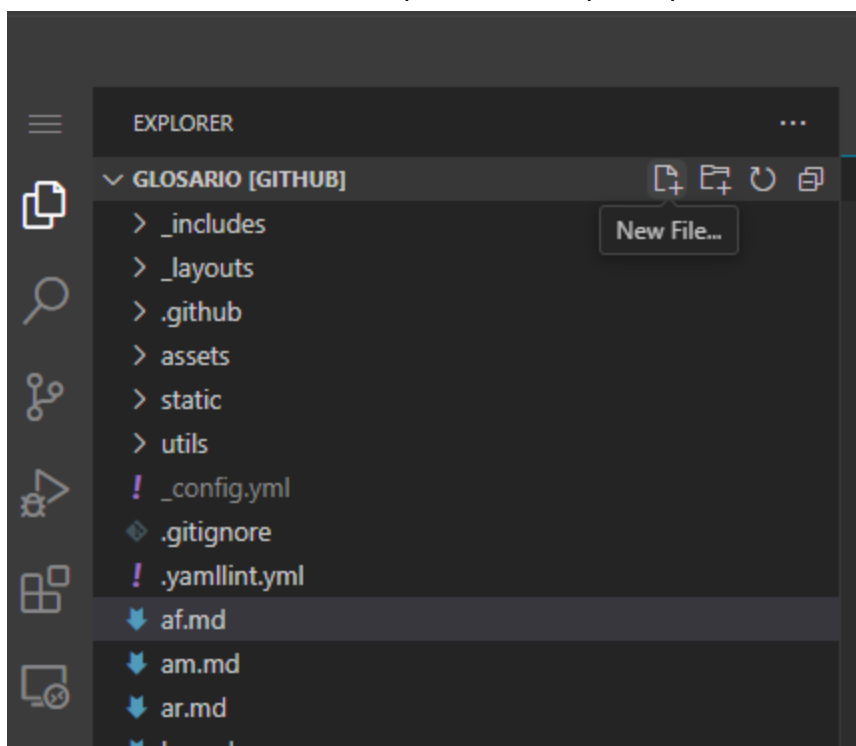
12. After you submit your **pull request**, the Glosario editors and reviewers will review your submission. Once the review process has been completed, your submission will be merged into the *glossary.yml* file.

Adding a new language

If you want to provide translations for a language that is currently not supported, you need to create a new Markdown template file and provide a corresponding **blurb** translation.

Adding a new language template file

1. Sign in to GitHub and go to the Glosario **repository** (<https://github.com/carpentries/glosario>)
2. Hit the '.' (period) key on your keyboard to open the GitHub dev environment in your browser.
3. Scrolling down in the file explorer window on the left-hand side of the screen, you will see **markdown (.md)** files corresponding to each language, e.g., fr.md for French, de.md for German, etc.
4. Click the New File link at the top of the file explorer pane.



- 5.
6. In the new box in the file list, type the two-letter language code of your chosen new language followed by the .md extension, and hit Enter.
7. Once created, double-click on the new file to start editing it
8. Add the following configuration within your file, replacing:
 - a. the `permalink` option with your two-letter language code prefixed and suffixed by slashes, e.g. `/af/`
 - b. the `direction` option with either `ltr` (for left-to-right) or `rtl` (right-to-left)
 - c. Leave all other text unchanged.

```

permalink: /<two-letter language code>/
layout: glossary
direction: <direction>
---
{% include glossary.html %}

```

9. If you need to, check how another language .md file is configured to check that yours looks correct.
10. Once complete, now check that a blurb translation exists for your chosen language by following the next section

Adding a new blurb translation

1. Following the creation of a new language file, a corresponding entry in the `_config.yml` file needs to exist to translate the Glosario blurb into that language
2. From the file explorer pane on the left, double-click on the `_config.yml` file to open it for editing.
3. Blurbs are added under the `languages` section, as per the Afrikaans (af) example below:

```

languages:
- key: af
  name: Afrikaans
  title: Woordelys
  blurb: >
    `glosario` is 'n oopbronwoordelys van terminologie wat in datawetenskap
    gebruik word en beskikbaar gestel word aanlyn sowel as 'n programmeer in
    [R] (https://github.com/carpentries/glosario-r/) en
    [Python] (https://github.com/carpentries/glosario-py/).
    Deur woordelyssleutels by 'n
    les se metadata in te sluit, kan outeurs aandui watter inligting in 'n les gedek
    word, wat leerders behoort te weet voordat die les begin en waar leerders
    die inligting kan vind. Outeurs kan ook die programmeerfunksies gebruik om
    konsekwente hiperskakels, vir terme en definisies in enige van verskeie tale,
    by hul lesse in te sluit.

```

4. Each `languages` entry has the following labels:
 - a. `key`: the two-letter language code of your language
 - b. `name`: the full name of your language
 - c. `title`: the translation of the word 'Glosario' in your language
 - d. `blurb`: > the translation of the blurb field text in the English language entry (en)
5. Scroll through the `_config.yml` file and find where your language code would be in alphabetical order.
6. Add in a new entry by supplying the key, name, title and blurb

7. Note that the `_config.yml` file format (**YAML**) needs a specific **indentation** to be used correctly, as follows:
 - a. The first indentation level (no spaces, aligned to the left-hand-side) is the `languages` section
 - b. The second indentation level (two spaces in, aligned to the first vertical line) is the key and language code
 - c. The third indentation level (four spaces in, aligned to the second vertical line) is the name and title labels, the blurb label, a colon, and a greater than an arrow, i.e. `blurb: >`
 - d. The fourth indentation level (six spaces in) is the blurb **translation definition**, where your translation should be added.
8. Once complete, please follow the next section to create a pull request

Creating the pull request

1. Once you have reviewed your language **markdown** file and/or `_config.yml` and checked the formatting against the other files, click the Source Control icon in the button menu on the far left of the screen (it should be underneath the magnifying glass icon and should have a circle with a number in it denoting the number of changes to **commit**).
2. Enter a message for the **commit** in the box in the top left, e.g., 'Add new language <two-letter language code>.'
3. Click **Commit** and **Push**.
4. You need to exit the **codespace** environment now. To do so, click the three-line **hamburger** icon at the top left to access the main menu.
 - a. Click on *"Go to Repository."*
 - b. You will now see your **pull request**; click *"Compare & pull request."*
 - c. Add a title for the pull request; in this case, the same as the **commit** message is OK - 'Add new language <two-letter language code>' and add a description. Add your GitHub handle and the language you have added. You can now click on *"Create pull request."*
5. Congratulations - you are done!