

# Lecture 13: Collaboration on Git/GitHub

LING 1340/2340: Data Science for Linguists

Na-Rae Han

# Objectives

---

## ▶ Git/GitHub

- ◆ Collaborating on the *same* repo – push access

## ▶ Wrap up: ML intros

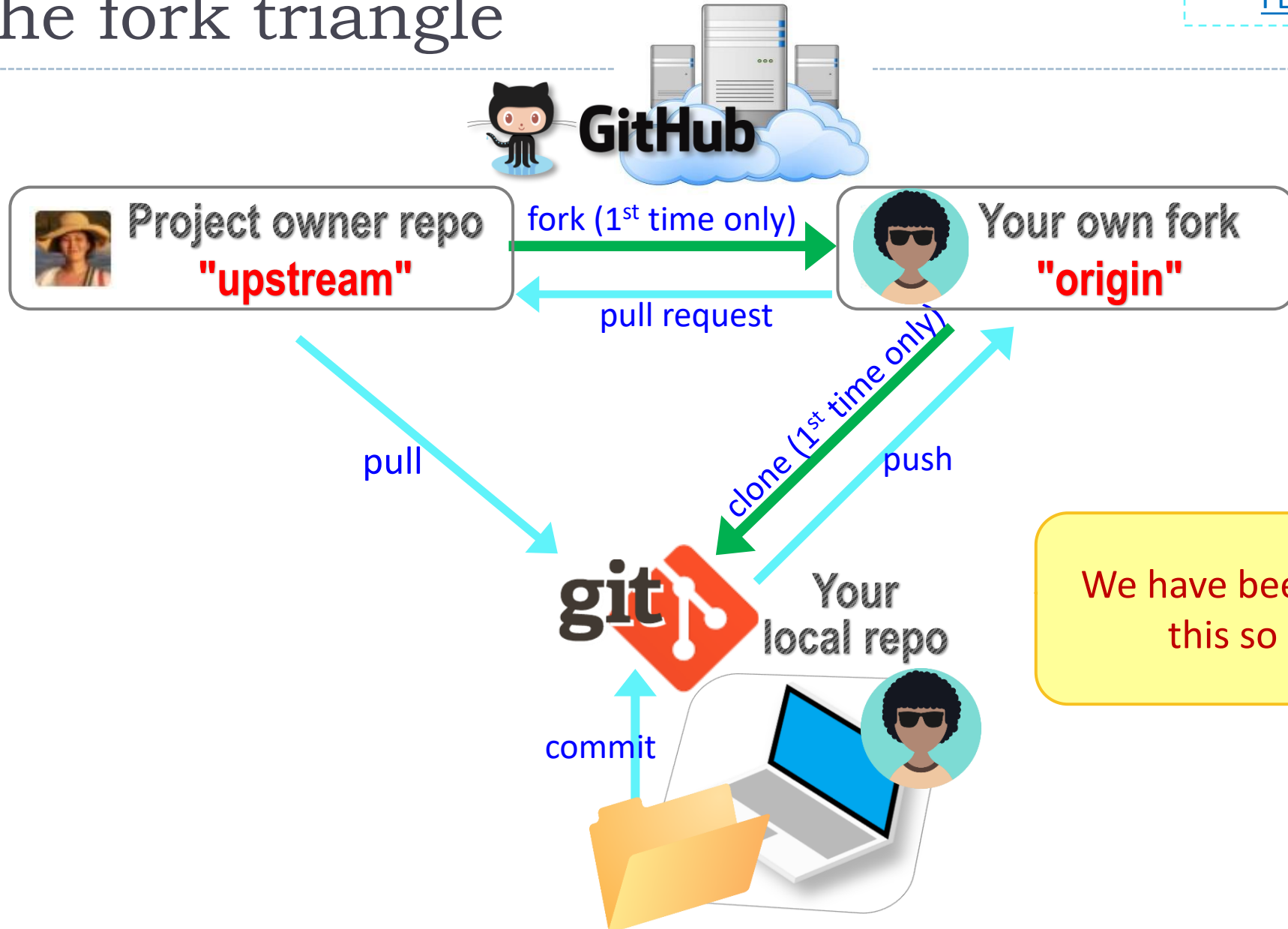
- ◆ Cross-validation results (JNB3)
- ◆ Feature weights in Naïve Bayes Model (JNB2)

## ▶ HW #3 review

- ◆ Task 2 prompt identification

\*Avatar icons by [FLATICON](https://flaticon.com)

# The fork triangle

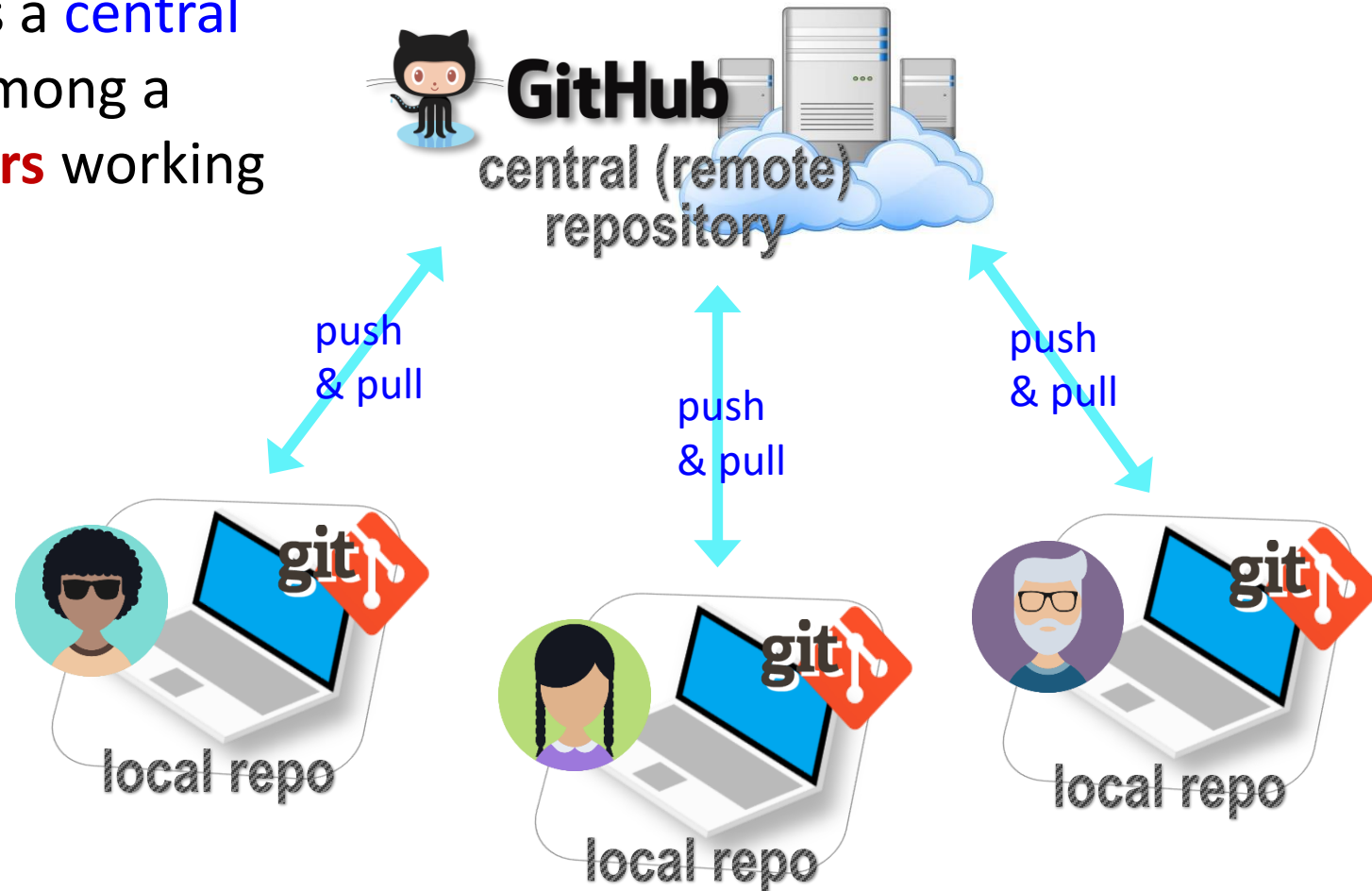


We have been doing this so far.

# GitHub: a *social*, remote repository

- ▶ GitHub also works as a **central remote repository** among a group of **collaborators** working on a shared project.

- ◆ Everyone works on their own *local* copy of the repository, making changes.
- ◆ Git is able to keep track and merge changes submitted by everyone.
- ◆ Everyone is an **equal collaborator** with push (=write) access.



**We are now ready!**

# Introducing... "Class-Lounge"

▶ <https://github.com/Data-Science-for-Linguists-2024/Class-Lounge>

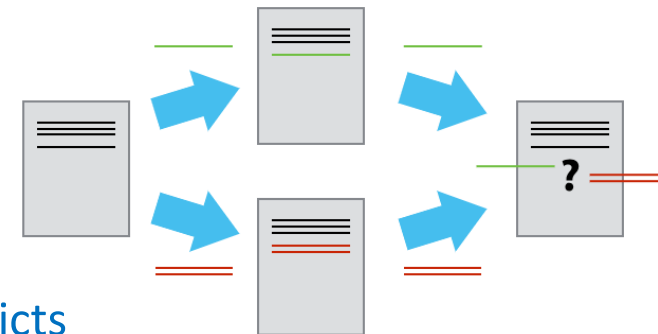
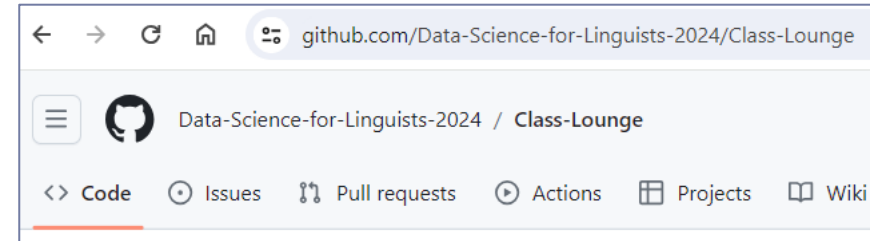
▶ Public!

▶ Everyone is listed as a "collaborator".

- ◆ Meaning, everyone has push access.
- ◆ No need to fork: pull and push directly.
- ◆ We will also truly collaborate: **edit shared files**.

▶ This means: **CONFLICTS**

- ◆ Na-Rae's tutorial on Git conflicts:
  - ◆ [https://github.com/mcdonn/LSA2019-Reproducible-Research/blob/master/linking\\_git\\_and\\_github.md#conflicts](https://github.com/mcdonn/LSA2019-Reproducible-Research/blob/master/linking_git_and_github.md#conflicts)



# When there is a conflict



- ▶ After you pull, Git **changes your file content**, which now looks like:

```
<<<<<<< HEAD
There was copyright information here.

=====
"RTlexdec", "RTnaming", "Familiarity", "Word", "AgeSubject", "WordC
tenFrequency", "WrittenSpokenFrequencyRatio", "FamilySize", "Deri
y", "InflectionalEntropy", "NumberSimplexSynsets", "NumberComplex
thInLetters", "Ncount", "MeanBigramFrequency", "FrequencyInitialD
>>>>>> c954b23f86b629c569223e2c0c38e32a0d870d22
```

You must **manually edit the file** and tidy up these lines.  
(aka resolve conflict)

# A GitHub race: our favorite animals

---

Activity  
10 minutes



1. Everyone was already added to the repo as a collaborator.
2. Clone the repo to your laptop.
3. Edit "favorite\_animals.md", add your line.
4. Do your usual local git routine: adding, committing.
5. Try pushing. It is likely you run into a conflict (someone else pushed in the meantime) and Git tells you to pull first.
6. Pull to receive the new updates.
  - ◆ Pay attention to the Git log message! It tells you how the file was changed.
7. Open "favorite\_animals.md", edit to resolve conflict.
8. Go back to step 4. Hope you were quick enough this time!

# Wrapping up

---

- ▶ HW #3 complete HW due on Wed!