Shared task: Git and Gtihub

This section explains how we use git and github for the tasks we solve during the shared task. The logistic is following:

- all students form a *github organization*
- teams will be registered as *github teams* as a part of the organization
- each team will have its own *git/github code repository* to synchronize a codebase with a server and among each other
- each team will have its own *github project board* to manage project tasks

Working with github project board

In order to open *project board* go to the project repository and click *project* tab. Here you can organize your work in Kanban-flow style:

- ❖ You have 3 main boards: To Do, In Progress, and Done. You can create and move *tasks* between them.
  - ➢ Example tasks: "Tokenize data", "Perform BPE split", "Translate test set", "Add visualization to the training process", etc
- ❖ Convert *task* to *issue* if you have some question or want to discuss it. Always convert task to issue, and assign it to yourself, if u are going to work on it.
- ❖ You will have several *github milstones* – shared task deadlines. You should connect issues to the corresponding milestones.

All the milestones are created for you.

You can find more about github project board here.

Working with git code repository

If you are not familiar with SVC at all, please find 10 minutes to get some basics (see link in the end of document). The essential commands are:

- Add changes you made to be ready to commit
- Commit – fix changes in your local repository (codebase).
- Push – move commit to github
- Pull – merge remote codebase with your local
- Status – check what files are changed and added.

You can use these command in following way:

1. Pull all the changes to your local repo
2. Make a code change on your laptop, add, and commit it
3. Push it to your team's shared github repo
4. SSH to the server, and pull all the changes from github
5. Execute recent version of the code on server with SLURM

However, it is not necessary, and you can do all the changes on server from the command line and just push them to the remote github repository.

NB: to avoid merge problems during pushes and pulls, each team member should work on different code file at the same time.

Project structure

<u>OpenNMT-py</u> – contains NMT framework to use

<u>Data</u> – contains text corpora both raw and preprocessed

<u>Models</u> – contains serialized OpenNMT-py models

<u>Reports</u> – contains milestone reports of your team

Feel free to expand and build on to of this folder structure.

Communicating results

Before each milstone deadline, make a special commit called "MILSTONE REPORT: <milstone name>" and push it to the remote github repo ("reports" folder).

Starting on repository

Use *git clone* command to download your repository.

Also, please specify your e-main and username, so we see who makes commits:

```
git config user.email "email@example.com"
git config user.name "John Doe"
```

Lastly, copy raw data to your *data* folder as specified in the end of the lab1.pdf.

Links

Github docs: https://help.github.com/

Git basics: https://guides.github.com/activities/hello-world/