How to use *GitHub*

Instructor: Dr. Jintao Ke, Spring 2024

1 Introduction

GitHub is a web-based platform for version control and collaboration that helps developers and teams to manage and track changes to their code repositories. It allows users to store their code in remote repositories and provides tools for code review, project management, and team collaboration. GitHub is widely used in the software development community and has become an essential tool for open-source development.

In this tutorial, you will learn the basic information of GitHub, how to download the code and how to interace with the author.

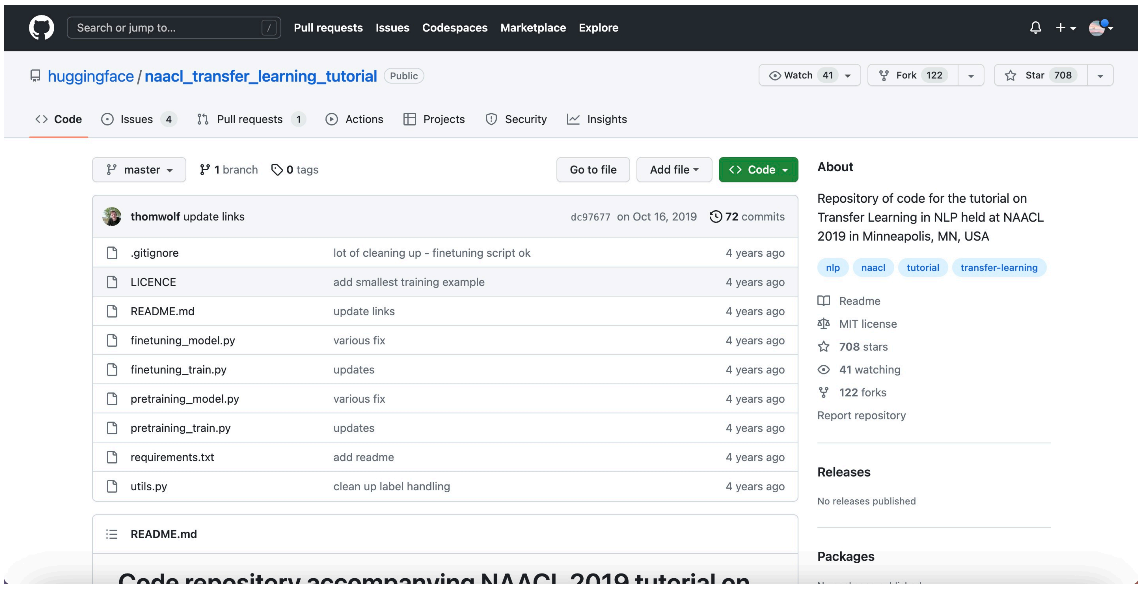
2 Basic Uasge of GitHub

2.1 Getting started with the code

Let's getting started with an example.

This is a GitHub link: https://github.com/huggingface/naacl_transfer_learning_tutorial

When you click the link, you will see a project (also called *repository*) page like Figure 1:



thomwolf update links — dc97677 on Oct 16, 2019 — 72 commits

.gitignore	lot of cleaning up - finetuning script ok	4 years ago
LICENCE	add smallest training example	4 years ago
README.md	update links	4 years ago
finetuning_model.py	various fix	4 years ago
finetuning_train.py	updates	4 years ago
pretraining_model.py	various fix	4 years ago
pretraining_train.py	updates	4 years ago
requirements.txt	add readme	4 years ago
utils.py	clean up label handling	4 years ago

README.md

Code repository accompanying NAACL 2019 tutorial on

About

Repository of code for the tutorial on Transfer Learning in NLP held at NAACL 2019 in Minneapolis, MN, USA

nlp naacl tutorial transfer-learning

Readme
MIT license
708 stars
41 watching
122 forks
Report repository

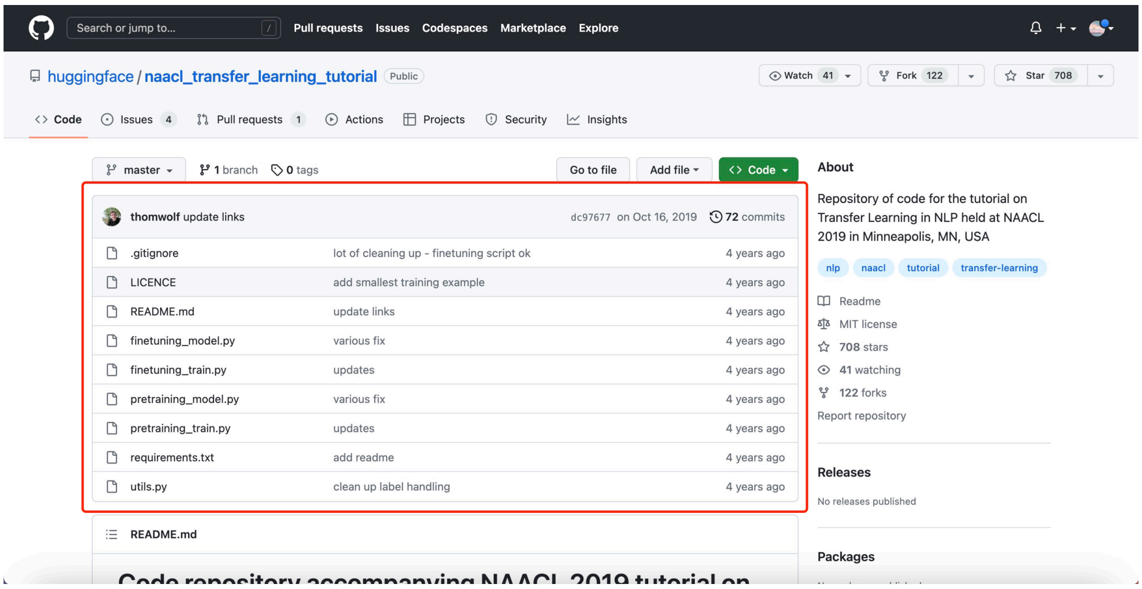
Releases

No releases published

Packages

Figure 1: GitHub Project Page

All the source code can be found here in the red box as Figure 2 shows:



thomwolf update links	dc97677 on Oct 16, 2019	72 commits
.gitignore	lot of cleaning up - finetuning script ok	4 years ago
LICENCE	add smallest training example	4 years ago
README.md	update links	4 years ago
finetuning_model.py	various fix	4 years ago
finetuning_train.py	updates	4 years ago
pretraining_model.py	various fix	4 years ago
pretraining_train.py	updates	4 years ago
requirements.txt	add readme	4 years ago
utils.py	clean up label handling	4 years ago

About

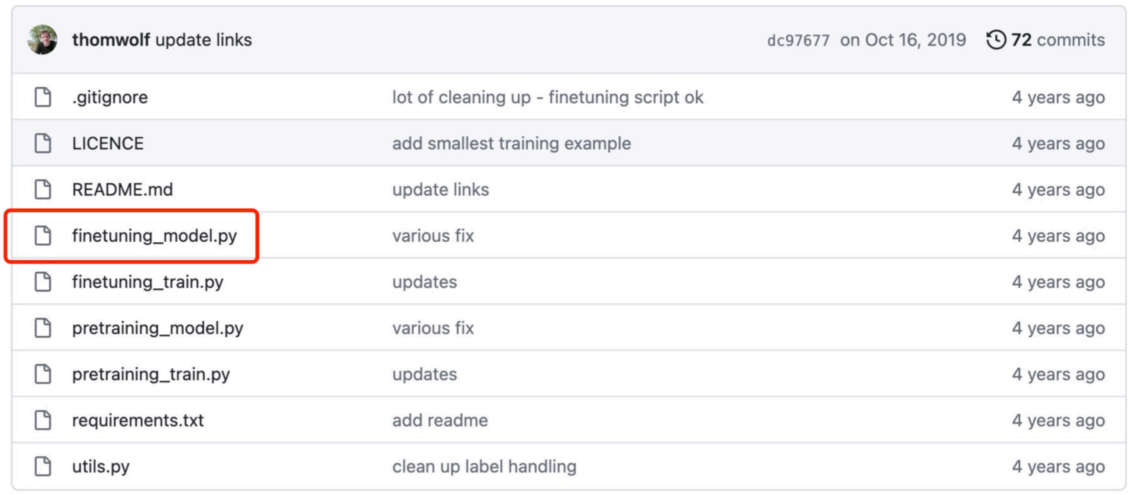
Repository of code for the tutorial on Transfer Learning in NLP held at NAACL 2019 in Minneapolis, MN, USA

nlp naacl tutorial transfer-learning

Readme
MIT license
708 stars
41 watching
122 forks
Report repository

Figure 2: Source Code

You can click one of the files to view the content like Figure 3&4 shows:

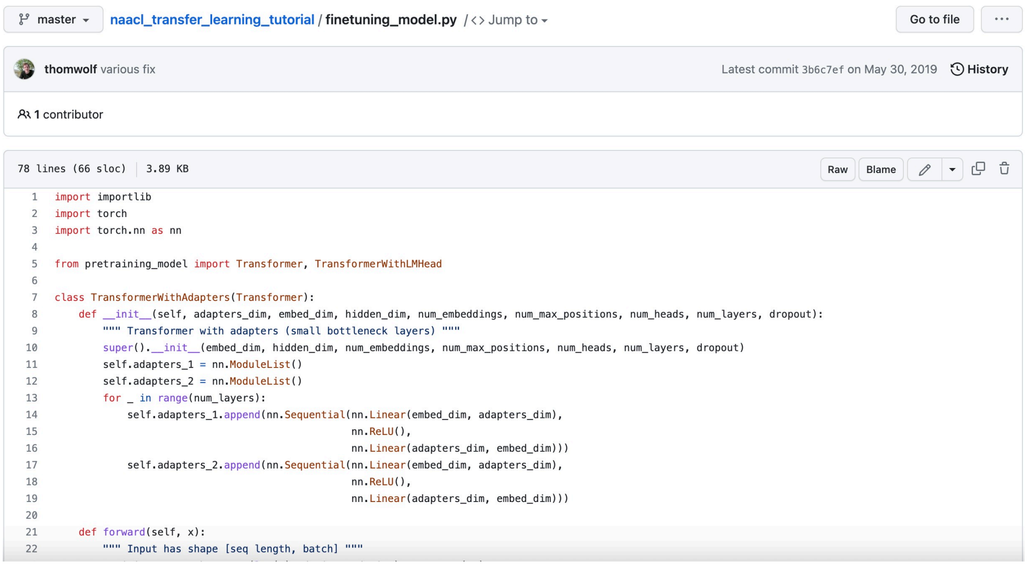


thomwolf update links	dc97677 on Oct 16, 2019	72 commits
.gitignore	lot of cleaning up - finetuning script ok	4 years ago
LICENCE	add smallest training example	4 years ago
README.md	update links	4 years ago
finetuning_model.py	various fix	4 years ago
finetuning_train.py	updates	4 years ago
pretraining_model.py	various fix	4 years ago
pretraining_train.py	updates	4 years ago
requirements.txt	add readme	4 years ago
utils.py	clean up label handling	4 years ago

Figure 3: Select a File

Let's turn back to the whloe project and there is a module named **README.md**[1] below the source code as Figure 5 shows. The README module interprets how you can run the whole project and reminds you of relevant precautions. Generally, if you are intended to use some code from GitHub, detailedly reading the README module can be important especially when you are not familiar with the content.

[1].md file is a document written in **Markdown**, a lightweight markup language, which is easy to read and write, and supports graphics, graphs, and math. You don't have to know it for this course, but if you aim to pursue a professional education or career in programming field, it is highly recommended that you learn the markdown language for better communication. This is a link for the introduction of markdown.


master
naacl_transfer_learning_tutorial / finetuning_model.py / <> Jump to
Go to file
thomwolf various fix
Latest commit 3b6c7ef on May 30, 2019
History
1 contributor
Raw
Blame

```
78 lines (66 sloc) | 3.89 KB
```

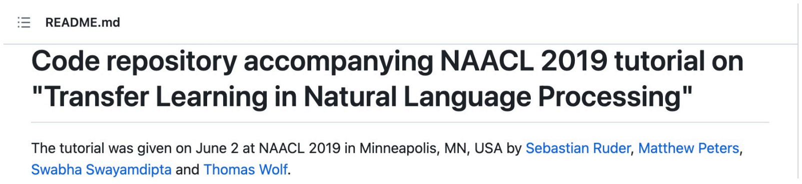
```python
import importlib
import torch
import torch.nn as nn

from pretraining_model import Transformer, TransformerWithLMHead

class TransformerWithAdapters(Transformer):
    def __init__(self, adapters_dim, embed_dim, hidden_dim, num_embeddings, num_max_positions, num_heads, num_layers, dropout):
        """ Transformer with adapters (small bottleneck layers) """
        super().__init__(embed_dim, hidden_dim, num_embeddings, num_max_positions, num_heads, num_layers, dropout)
        self.adapters_1 = nn.ModuleList()
        self.adapters_2 = nn.ModuleList()
        for _ in range(num_layers):
            self.adapters_1.append(nn.Sequential(nn.Linear(embed_dim, adapters_dim),
                                                 nn.ReLU(),
                                                 nn.Linear(adapters_dim, embed_dim)))
            self.adapters_2.append(nn.Sequential(nn.Linear(embed_dim, adapters_dim),
                                                 nn.ReLU(),
                                                 nn.Linear(adapters_dim, embed_dim)))

    def forward(self, x):
        """ Input has shape [seq length, batch] """
```

Figure 4: The Detailed Content


README.md

Code repository accompanying NAACL 2019 tutorial on "Transfer Learning in Natural Language Processing"

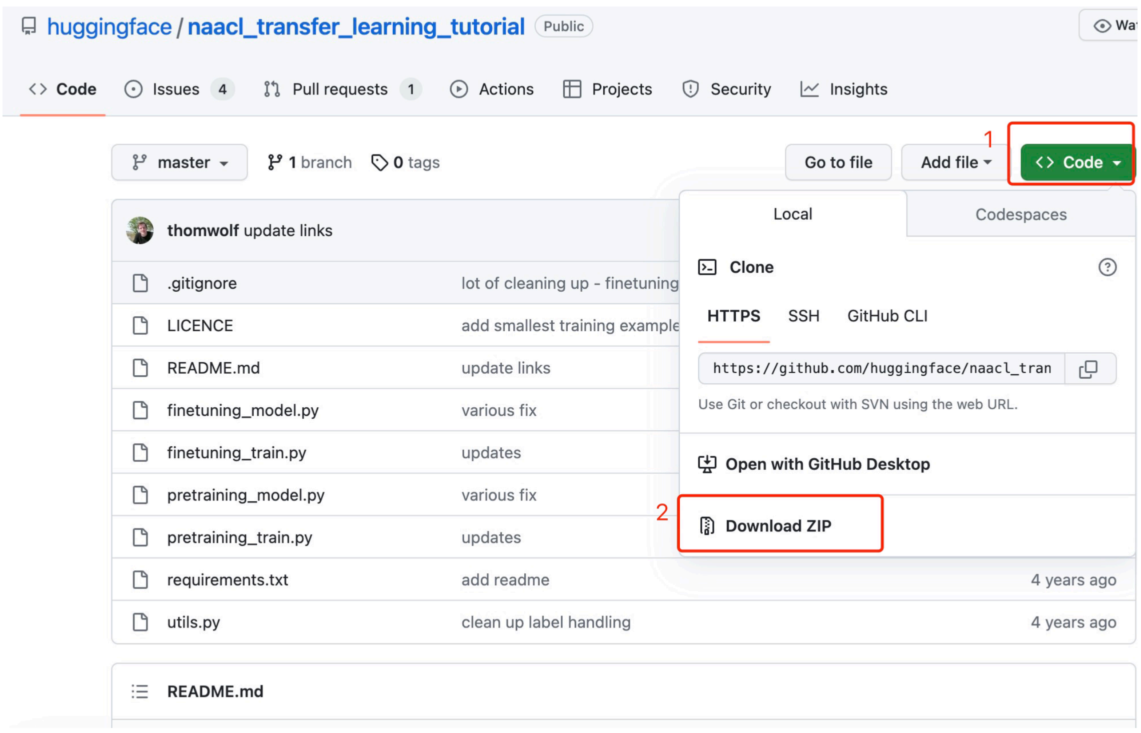
The tutorial was given on June 2 at NAACL 2019 in Minneapolis, MN, USA by Sebastian Ruder, Matthew Peters, Swabha Swayamdipta and Thomas Wolf.

Figure 5: README.md

After familiarizing yourself with the whole project, you might want to deploy the code in your personal computer. You don't have to copy the code for each file, because GitHub offers you a download button to get a zip file contaning the whole project as Figure 6 shows:

Click the *Code* button of green color and then click the *Download ZIP* button, you will get a zip file like the Figure 7 shows:

Finally, decompress the downloaded compressed file and you can start the project as the Figure 8 shows:



https://github.com/huggingface/naacl_tran

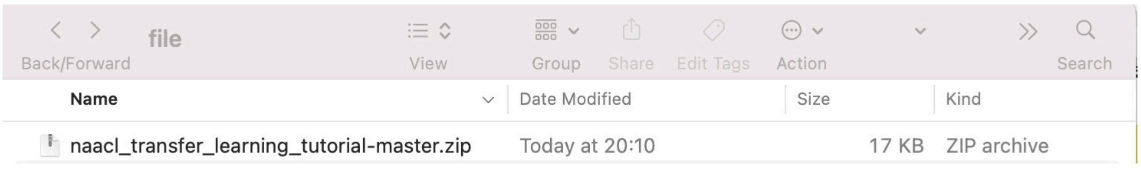
Use Git or checkout with SVN using the web URL.

thomwolf update links

.gitignore	lot of cleaning up - finetuning	
LICENCE	add smallest training example	
README.md	update links	
finetuning_model.py	various fix	
finetuning_train.py	updates	
pretraining_model.py	various fix	
pretraining_train.py	updates	
requirements.txt	add readme	4 years ago
utils.py	clean up label handling	4 years ago

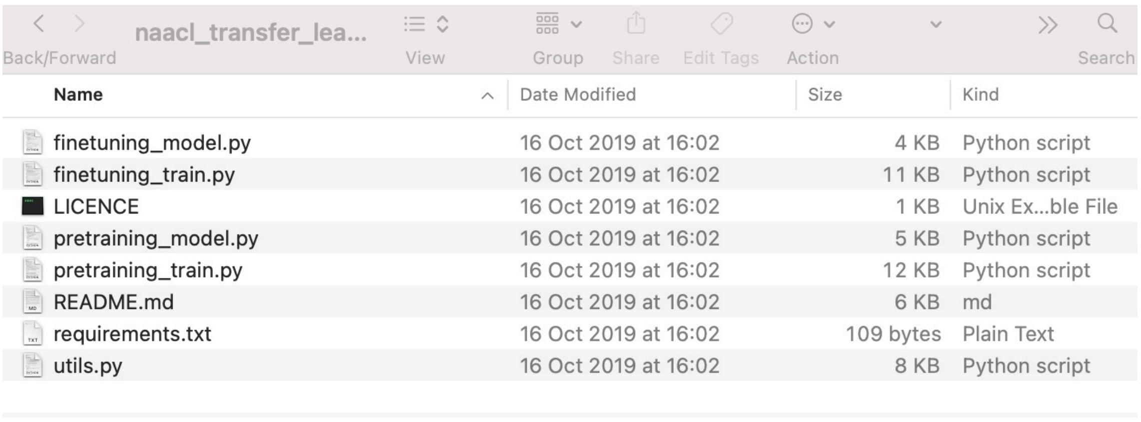
README.md

Figure 6: Download the Zip File



Name	Date Modified	Size	Kind
naacl_transfer_learning_tutorial-master.zip	Today at 20:10	17 KB	ZIP archive

Figure 7: Zip File

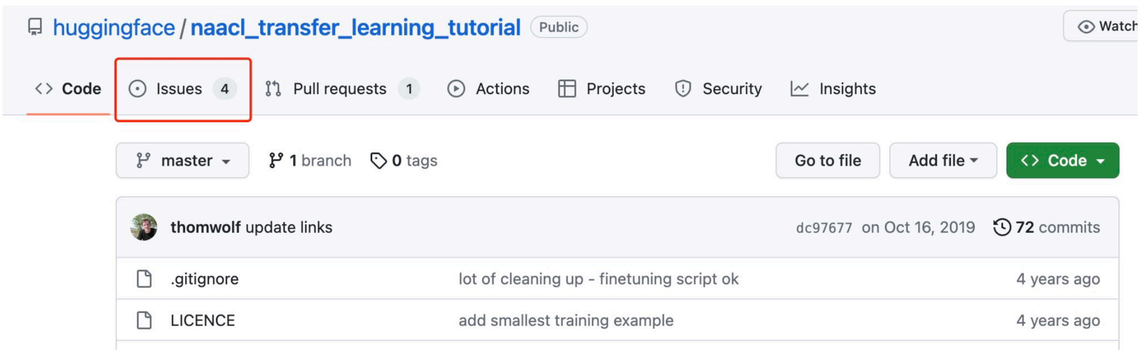


Name	Date Modified	Size	Kind
finetuning_model.py	16 Oct 2019 at 16:02	4 KB	Python script
finetuning_train.py	16 Oct 2019 at 16:02	11 KB	Python script
LICENCE	16 Oct 2019 at 16:02	1 KB	Unix Ex...ble File
pretraining_model.py	16 Oct 2019 at 16:02	5 KB	Python script
pretraining_train.py	16 Oct 2019 at 16:02	12 KB	Python script
README.md	16 Oct 2019 at 16:02	6 KB	md
requirements.txt	16 Oct 2019 at 16:02	109 bytes	Plain Text
utils.py	16 Oct 2019 at 16:02	8 KB	Python script

Figure 8: The Decompressed Project

2.2 How to communicate with the project author

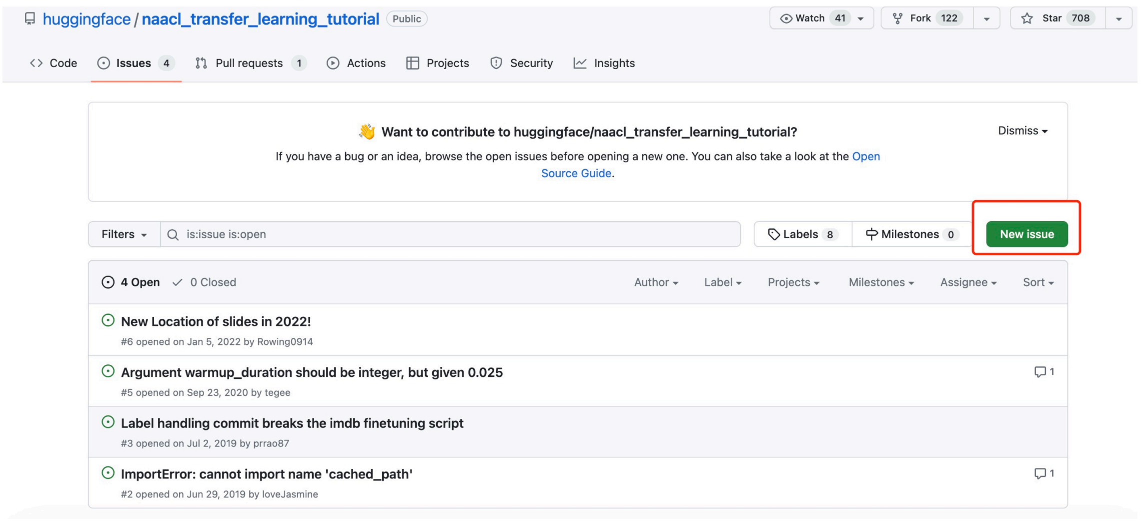
In a GitHub repository, you can leave a comment to the author in the project. As the Figure 9&10 shows, you can click the issue button and find the existing issues in the project.



thomwolf update links dc97677 on Oct 16, 2019 72 commits

.gitignore	lot of cleaning up - finetuning script ok	4 years ago
LICENCE	add smallest training example	4 years ago

Figure 9: Issue Button



👋 Want to contribute to huggingface/naacl_transfer_learning_tutorial?

If you have a bug or an idea, browse the open issues before opening a new one. You can also take a look at the Open Source Guide.

New Location of slides in 2022!
#6 opened on Jan 5, 2022 by Rowing0914

Argument warmup_duration should be integer, but given 0.025 — 1
#5 opened on Sep 23, 2020 by tegee

Label handling commit breaks the imdb finetuning script
#3 opened on Jul 2, 2019 by prrao87

ImportError: cannot import name 'cached_path' — 1
#2 opened on Jun 29, 2019 by loveJasmine

Figure 10: Issues in a Project

You can click one of the existing issues to see some problems discussed by others or create a new issue by clicking the *New issue* button. If doing so, the project author (TA) can notice your comment and reply to your questions.

3 Git

3.1 Key Concepts

Git is a distributed version control system that is widely used for managing and tracking changes in software development projects, which allows multiple people to collaborate on the same codebase efficiently while keeping track of changes, history, and different versions of the code.

Some Key Concepts of Git are helpful for you to speed quickly with the relevant content:

1. **Repositories**: A repository (repo) in Git is a storage unit that holds all the files, history, and configurations related to a project. Repositories can be hosted on remote servers (e.g., GitHub, GitLab) or locally on your computer.
2. **Commits**: A commit represents a snapshot of changes to the code at a specific point in time. Each commit has a unique identifier (SHA-1 hash) and contains information about the changes made and who made them.

3. **Branches**: Branches allow developers to work on different features or bug fixes concurrently without interfering with each other's work. Each branch is an independent line of development.

4. **Merging and Pull Requests**: Merging combines changes from one branch into another, often used to integrate feature branches into the main development branch. Pull requests (or merge requests) facilitate code review and discussion before merging changes.

5. **Remote Repositories**: Remote repositories are hosted on servers accessible over the internet. Developers can collaborate by pushing their changes to a remote repo and pulling others' changes to their local repo.

6. **Clone, Pull, Push**:
   - **Clone**: Creating a local copy of a remote repository on your computer.
   - **Pull**: Fetching and integrating changes from a remote repository into your local repository.
   - **Push**: Uploading your local commits to a remote repository.

3.2 The mothods of Using Git

If you are using the computers with **MacOS**, including MacBook series (MacBook Air, MacBook Pro), iMac, Mac mini and Mac Pro, etc., you can use Git directly in the **Terminal**, which can be found in **Spotlight Search** or **Finder**.

If you are using the computers with **Windows** operating system, like desktop and laptop computers, you will need to install a unix-like shell environment on Windows, making it convenient to use Git commands. **Git Bash** is a command-line interface for Git on Windows and in order to download and install Git Bash, go to https://git-scm.com/downloads and click on the "Download for Windows" button. This will download the Git for Windows installer.

3.3 Start With Git

1. Open the Git environment: for MacOS users, just open the **Terminal**, and for Windows users, you need to open the **Git Bash**.

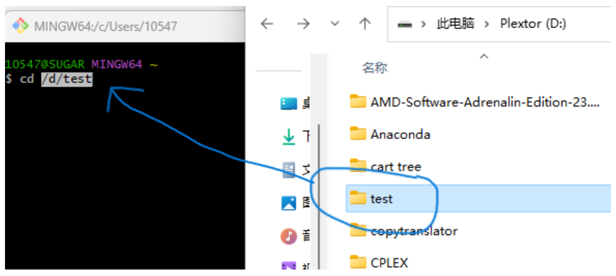
2. Set the username and email (–global is a global parameter, indicating that this configuration will be used for all local Git repositories). Use the command line below:

```
git config --global user.name "yourname"

git config --global user.email "your_email@youremail.com"
```

Listing 1: Command Line to set the username and email

3. You need to use '**cd**' to change your path to the desired location. For example, here I'm using "test." First, type cd in the command line, press the spacebar, then drag and drop the folder with the desired path into the command line as the Figure 11 shows.



```
10547@SUGAR MINGW64 ~
$ cd /d/test
```

Figure 11: Change your path

So the command is:

```
cd "your path"
```

Listing 2: Command Line to change the path

4. Use **git clone** to create a copy of a remote repository on your local machine, meaning that you obtain a full copy of all the files, commit history, and branches from the remote repository. So after doing this, you will get all the code and data you need. In this course, you will using the following command line to get the tutorials and exercise:

```
git clone https://github.com/hku-kejintao/HKU-CIVL3140-AI-in-Civil-
    Engineering.git"
```

Listing 3: Command Line to clone the repository

5. After clonging the repository, you have got all the necessary code and data, but sometimes there might be some changes for the tutorials or exercise. To keep the local code and data synchronized with the GitHub repository, you can use **remote add** and **pull** request:

```
git remote add upstream https://github.com/hku-kejintao/HKU-CIVL3140-AI-in-
    Civil-Engineering.git

git pull upstream main
```

Listing 4: Command Line to add remote and pull the repository

3.4 Git Push

If you are intended to commit some changes to a remote repository, you may need **git push** to do it. However, this **git push** operation can only work with the repository you own or authorized by others. The following procedures are the instructions that how to use **git push** operation:

1. **Generate SSH Key**: Open a terminal window and enter the following command to generate an SSH key pair. Replace <your_email@example.com> with your email address.

```
ssh-keygen -t rsa -b 4096 -C "<your_email@example.com>"
```

Listing 5: generate an SSH key pair

Press Enter to proceed. You can save the key in a specific location by adding -f /path/to/key.

2. **Accessing Your Public Key**: Your public key is saved in the /.ssh/id_rsa.pub file. Use a text editor to view and copy it.

3. **Adding SSH Key to GitHub**: Sign in to GitHub and log in to your GitHub account. Click on your profile picture in the top right corner, then select "Settings." In the left sidebar, click on "SSH and GPG keys." Click on "New SSH key." Give your key a title and paste your copied public key into the "Key" field. Click on "Add SSH key" to save.

4. **Add remote**: use **git remote add origin** to connect your local repository to remote one.

```
git remote add origin <repository_url>
```

Listing 6: git remote

5. **Use git push**: in a git command environment, make sure you have clone a repository and change your path right. If you have made some changes and want to commit them, you can use the following commands (change the 'comment' with your own annotations):

```
        git add .

        git commit -m "comment"

        git push origin <branch_name>
```

Listing 7: git push

Replace <branch_name> with the branch you want to push to (often "main").