How to import Python packages from GitHub into a Colab

Have you ever tried to install a package from GitHub on your Google Colab? Some time ago, I had this same question. With a little research, I found the solution, which I share with you in this post. And, believe me, importing a Python package from GitHub is easier than you think!

Why is it so important?

Learning how to import a package may seem a secondary task, but sometimes you will want to use a module or something you created on a Tensorflow project.

You can upload documents and other stuff using the Colab interface. But if you stop a session, you will have to manually upload everything back.

A simpler way of uploading things is cloning a GitHub repository and using command line to automate this step.

Import and install

First thing to do is cloning the repository you want to use.

Then use **%cd** to navigate to the directory you want.

If you have run the clone command before, you will see a message saying that the files are already downloaded. You can restart your session (everything will be erased) or use the pull command to get the changes made on the repository before installing the requirements and the package itself.

The package I am trying to install depends on other packages to work. Let’s install these using the **!pip** command and the **requirements.txt** file.

Then, I will proceed to the install of the package itself. Use the command **!python setup.py install** to do it.

The last thing to do is importing your package!

```
# clone package repository
!git clone https://github.com/vallantin/atalaia.git
# navigate to atalaia directory
%cd atalaia
# get modifications made on the repo
!git pull origin master
# install packages requirements
!pip install -r requirements.txt
# install package
!python setup.py install
# import it
from atalaia.atalaia import Atalaia
```

Installing packages can have unexpected effects on your environment.

Different packages can have the same dependencies, but use different versions of them.

If you find problems, try to force/re-install the packages that are not working.

```
#!pip install --upgrade --force-reinstall **some_package_here**
```

Finally, import the other packages and pray for the best . Hopefully, you won’t spend one hour debugging as I did.

```
# do other imports to test if we have errors
import tensorflow as tf
import tensorflow_hub as hub
import tensorflow_datasets as tfds
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
import numpy as np
import pandas as pd
from pprint import pprint
```

Conclusion: what we learned today

Importing external packages is useful when you need to use something you developed locally.

Since not everyone has an external GPU at home, training Deep Learning models on Colab is very handy.