

Download Instructions

These are the instructions to download and run the code from the paper "A General Approach for Improving Deep Learning-based Medical Relation Extraction using a Pre-trained Model and Fine-tuning", by Tao Chen, Mingfen Wu and Hexi Li.

1. Request access to the N2B2 (previously called i2b2) corpus of annotated medical data from <https://portal.dbmi.hms.harvard.edu/>. Once your request is completed (this might take a few days), please download the corpus.
2. Download the Bert-Base model from the official website: <https://github.com/google-research/bert>. Please download using the highlighted link below, the other versions won't work with the code. Once downloaded, unzip the directory.

	H=128	H=256	H=512	H=768
L=2	2/128 (BERT-Tiny)	2/256	2/512	2/768
L=4	4/128	4/256 (BERT-Mini)	4/512 (BERT-Small)	4/768
L=6	6/128	6/256	6/512	6/768
L=8	8/128	8/256	8/512 (BERT-Medium)	8/768
L=10	10/128	10/256	10/512	10/768
L=12	12/128	12/256	12/512	12/768 (BERT-Base)

3. Clone the repository with the original code from <https://github.com/chentao1999/MedicalRelationExtraction> into your local computer. If you use the git command line, you can use the following command: "git clone <https://github.com/chentao1999/MedicalRelationExtraction.git>"
4. Open the directory on your computer where the code was downloaded. In the "pretrained bert model" subdirectory, paste the unzipped directory with the pre-trained BERT models

Name	Date modified	Type	Size
__pycache__	5/31/2020 11:38 AM	File folder	
bert	5/31/2020 11:38 AM	File folder	
corpus	5/31/2020 11:39 AM	File folder	
img	5/31/2020 11:39 AM	File folder	
pretrained_bert_model	5/31/2020 11:39 AM	File folder	
result	5/31/2020 11:39 AM	File folder	
data_process_BC5CDR.py	5/31/2020 10:59 AM	Python File	6 KB
data_process_i2b2.py	5/31/2020 10:59 AM	Python File	6 KB
data_process_TCM.py	5/31/2020 10:59 AM	Python File	6 KB
LICENSE	5/31/2020 10:59 AM	File	2 KB
README.md	5/31/2020 10:59 AM	MD File	4 KB
run_BC5CDR.py	5/31/2020 10:59 AM	Python File	11 KB
run_i2b2.py	5/31/2020 10:59 AM	Python File	12 KB
run_TCM.py	5/31/2020 10:59 AM	Python File	11 KB
text_model.py	5/31/2020 10:59 AM	Python File	12 KB

Medical > pretrained_bert_model			
Name	Date modified	Type	Size
uncased_L-12_H-768_A-12	5/31/2020 11:39 AM	File folder	
empty.txt	5/31/2020 11:04 AM	Text Document	1 KB

- Open the corpus/i2b2 subdirectory, and paste the annotations and the groundtruth folders you've downloaded from i2b2.

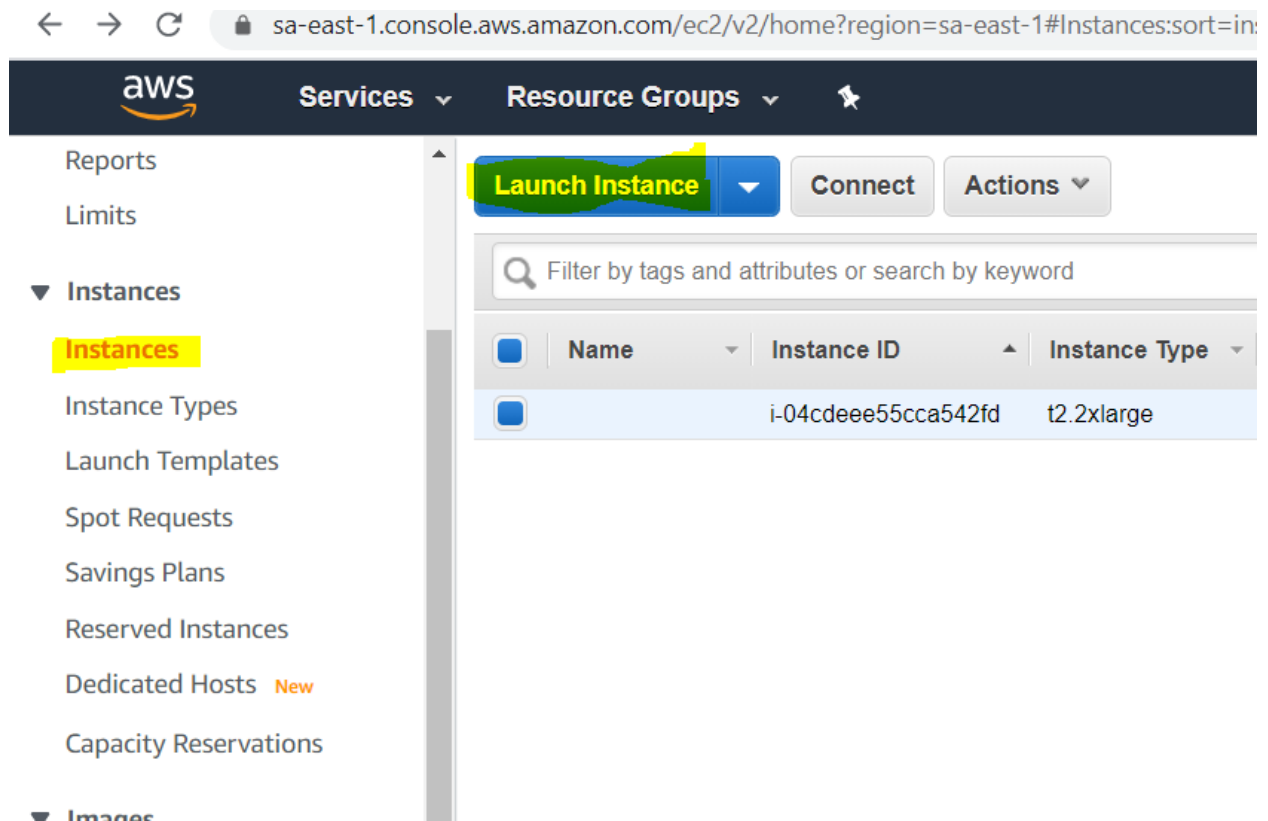
This PC > Desktop > Medical > corpus > i2b2			
Name	Date modified	Type	Size
2012-07-15.original-annotation.release	5/31/2020 11:39 AM	File folder	
ground_truth	5/31/2020 11:39 AM	File folder	
empty.txt	5/31/2020 10:59 AM	Text Document	1 KB
test.txt	5/31/2020 10:59 AM	Text Document	138,991 KB
train.txt	5/31/2020 10:59 AM	Text Document	134,883 KB

- Zip the root folder, the main directory that contains the code for this project. Your code is ready to be pasted into your virtual machine.

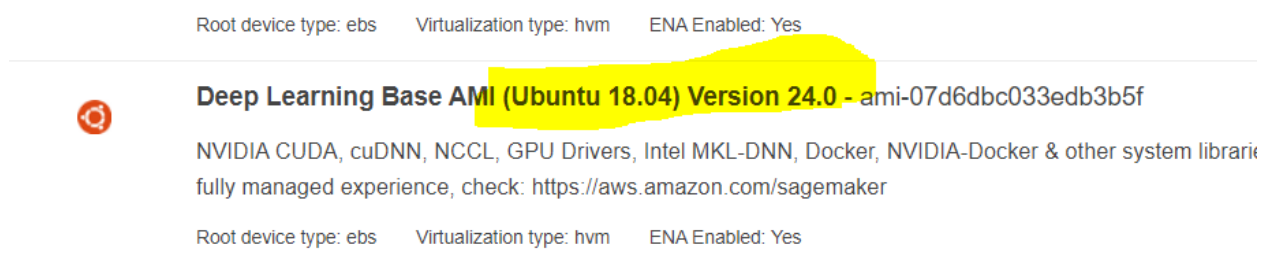
Setting up the machine

- For this project, I used Deep Learning Base AMI (Ubuntu 18.04) Version 24.0, with a t2.xlarge instance, running in São Paulo.

2. To create a new aws machine, go to the aws console and, on the right menu bar, select “Instances”. Then, select “Launch Instance”.

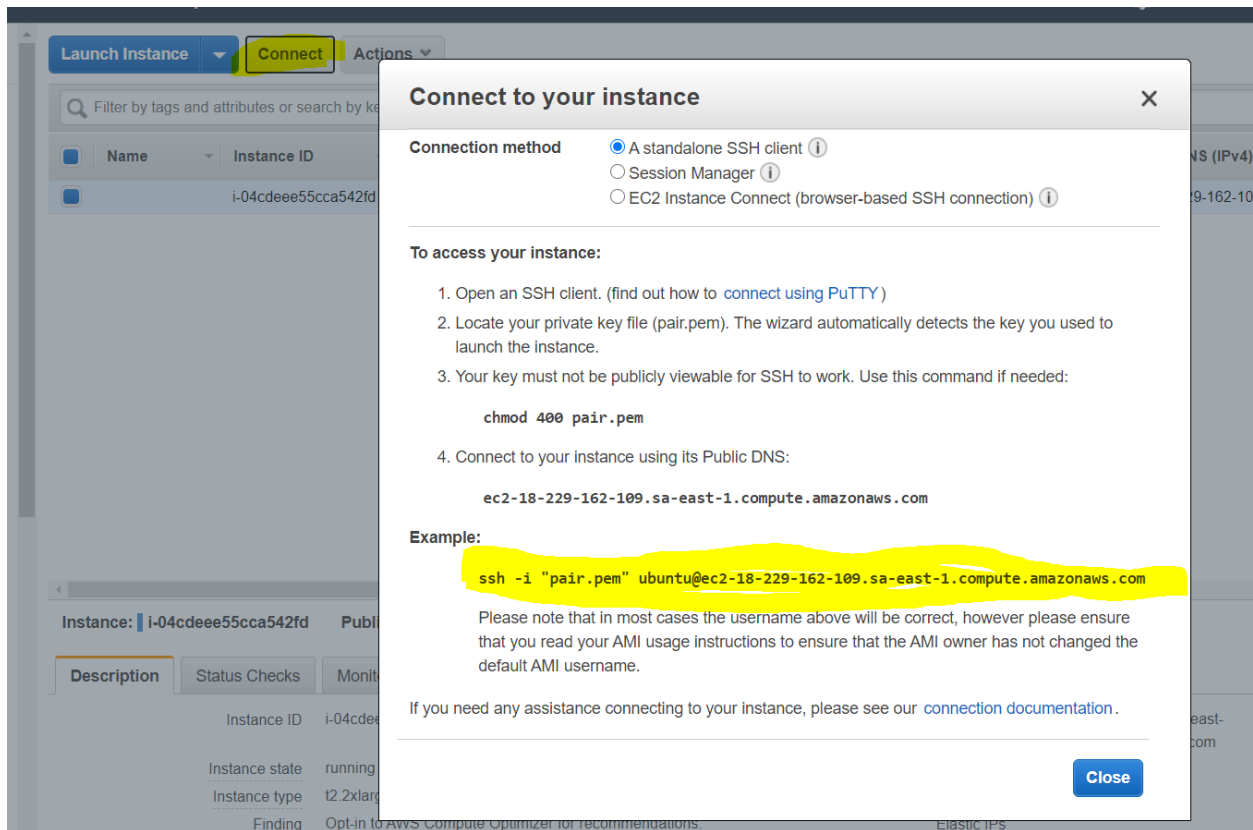


3. Type in “Sagemaker” in the search bar, and select the Deep Learning Base AMI (Ubuntu 18.04) Version 24.0.



4. Finish creating your virtual machine. For mine, I used the t2.2xlarge.
3. To connect to your machine, go to the aws console website, go to “instances”, select your instance and press connect. It will give you the ssh command you need to connect to your virtual machine using the command line. Please note

that you will need to check this command periodically: your public DNS will probably change whenever you restart the machine.



4. Still on your local computer, open a command prompt, cd to the directory where you keep your access key, paste and run the ssh command you've copied. If it asks you if you're sure if you want to connect, type 'yes'.

```
Command Prompt - ssh -i "pair.pem" ubuntu@ec2-18-229-162-109.sa-east-1.compute.amazonaws.com
Microsoft Windows [Version 10.0.18362.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Ana Flavia>cd Downloads

C:\Users\Ana Flavia\Downloads>ssh -i "pair.pem" ubuntu@ec2-18-229-162-109.sa-east-1.compute.amazonaws.com
The authenticity of host 'ec2-18-229-162-109.sa-east-1.compute.amazonaws.com (18.229.162.109)' can't be established.
ECDSA key fingerprint is SHA256:3FzXC9bUdA7zPh0ybzPxxRxhU2mMmPBSP2Z4Do2ozMU.
Are you sure you want to continue connecting (yes/no)?
```

5. If you see the information about the AMI, you've connected to the aws

```
Select ubuntu@ip-172-31-33-56: ~  
Last login: Wed Jun  3 22:31:18 2020 from 187.3.220.220  
=====
```

```
  _|_  _|_  )  
  _|_  _|_  /  Deep Learning Base AMI (Ubuntu 18.04) Version 24.0  
  _|_  _|_  |  
=====
```

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1019-aws x86_64v)  
  
Nvidia driver version: 440.33.01  
CUDA versions available: cuda-10.0 cuda-10.1 cuda-10.2 cuda-9.0 cuda-9.2  
Default CUDA version is 10.0  
Libraries: cuDNN, NCCL, Intel MKL-DNN  
  
AWS Deep Learning AMI Homepage: https://aws.amazon.com/machine-learning/amis/  
Developer Guide and Release Notes: https://docs.aws.amazon.com/dlami/latest/devguide/what-is-dlami.html  
Support: https://forums.aws.amazon.com/forum.jspa?forumID=263  
For a fully managed experience, check out Amazon SageMaker at https://aws.amazon.com/sagemaker  
When using INF1 type instances, please update regularly using the instructions at: https://github.com/aws/aws-neuron-sdk/tree/master/release-notes  
=====
```

```
* Documentation: https://help.ubuntu.com  
* Management:   https://landscape.canonical.com  
* Support:       https://ubuntu.com/advantage  
  
System information as of Mon Jun  8 18:30:29 UTC 2020  
  
System load:  0.0          Processes:            150  
Usage of /:   66.1% of 48.41GB  Users logged in:     0  
Memory usage: 0%          IP address for eth0: 172.31.33.56  
Swap usage:   0%          IP address for docker0: 172.17.0.1  
  
* MicroK8s gets a native Windows installer and command-line integration.  
  https://ubuntu.com/blog/microk8s-installers-windows-and-macos  
  
* Canonical Livepatch is available for installation.  
  - Reduce system reboots and improve kernel security. Activate at:  
    https://ubuntu.com/livepatch  
  
29 packages can be updated.  
0 updates are security updates.  
  
Last login: Wed Jun  3 22:31:18 2020 from 187.3.220.220  
ubuntu@ip-172-31-33-56:~$
```

6. Please run the following command: `pip3 install tensorflow==1.14`. This will downgrade your tensorflow (which is 2.0 or above) to the version of tensorflow used in the code.
7. Run the following code on the command line on your virtual machine: `jupyter notebook --port 8080`. This will create a Jupyter Notebook, which is a local website that allows us to see and access files. If this is your first time using the Jupyter notebook, it will also give you an access token. This will allow us to access the remote files from our local machine.

```
ubuntu@ip-172-31-33-56: ~  
ubuntu@ip-172-31-33-56:~$ jupyter notebook --port 8080  
[I 18:54:53.714 NotebookApp] Serving notebooks from local directory: /home/ubuntu  
[I 18:54:53.714 NotebookApp] The Jupyter Notebook is running at:  
[I 18:54:53.714 NotebookApp] http://localhost:8080/?token=6ad7568df76af8df5f92e183750393956b6731440ea8a63c  
[I 18:54:53.714 NotebookApp] or http://127.0.0.1:8080/?token=6ad7568df76af8df5f92e183750393956b6731440ea8a63c  
[I 18:54:53.714 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).  
[W 18:54:53.718 NotebookApp] No web browser found: could not locate runnable browser.  
[C 18:54:53.718 NotebookApp]  
  
To access the notebook, open this file in a browser:  
file:///home/ubuntu/.local/share/jupyter/runtime/nbserver-3043-open.html  
Or copy and paste one of these URLs:  
http://localhost:8080/?token=6ad7568df76af8df5f92e183750393956b6731440ea8a63c  
or http://127.0.0.1:8080/?token=6ad7568df76af8df5f92e183750393956b6731440ea8a63c
```

8. On your local computer, create a tunnel to the jupyter notebook: open a new command line window, go to the directory where you have your access key file to your aws machine, and run:

```
ssh -i "key.pem" -L 8080:localhost:8080 ubuntu@address
```

Where "key" is the name of your access key file and ubuntu@address is the username and address that you use to connect to your aws machine. For example, [ubuntu@ec2-54-235-229-89.compute-1.amazonaws.com](https://aws.amazon.com/ec2/instance-profiles/).

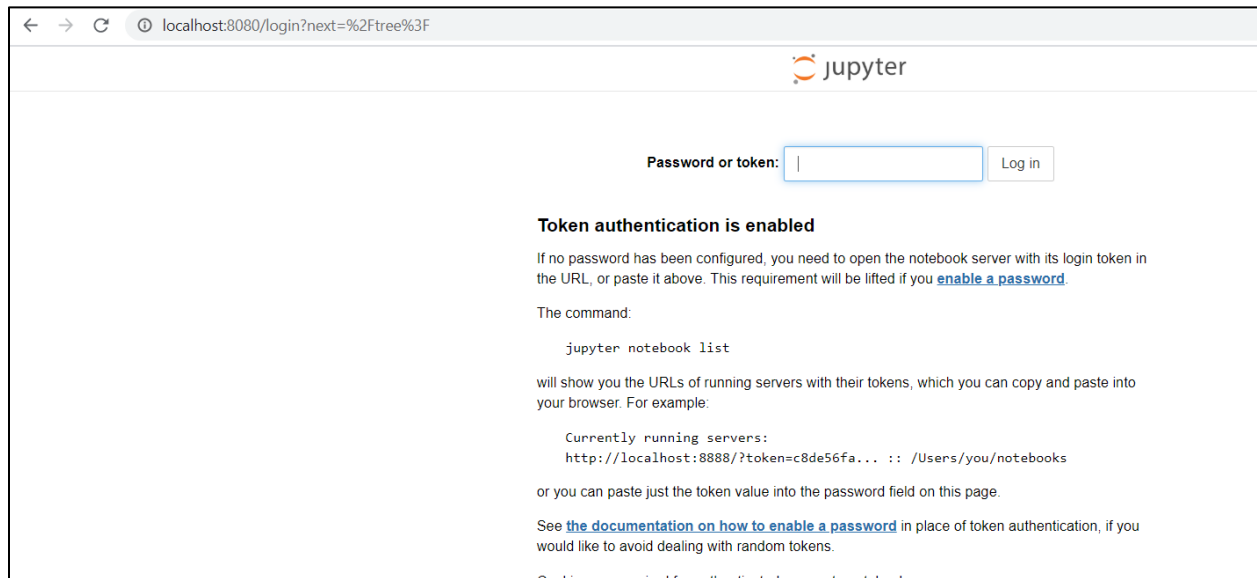
```
Command Prompt  
C:\Users\Ana Flavia>ssh -i "pair.pem" -L 8080:localhost:8080 ubuntu@ec2-18-229-162-109.sa-east-1.compute.amazonaws.com
```

```
ubuntu@ip-172-31-33-56: ~  
C:\Users\Ana Flavia>ssh -i "pair.pem" -L 8080:localhost:8080 ubuntu@ec2-18-229-162-109.sa-east-1.compute.amazonaws.com  
=====
```

```
  _ | _ | _ )  
  _ | ( _ /  Deep Learning Base AMI (Ubuntu 18.04) Version 24.0  
  _ | \ | _ |  
=====
```

```
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1019-aws x86_64v)  
  
Nvidia driver version: 440.33.01  
CUDA versions available: cuda-10.0 cuda-10.1 cuda-10.2 cuda-9.0 cuda-9.2  
Default CUDA version is 10.0  
Libraries: cuDNN, NCCL, Intel MKL-DNN  
  
AWS Deep Learning AMI Homepage: https://aws.amazon.com/machine-learning/amis/
```

1. On your local computer, using Google Chrome, go to localhost:8080. This is the jupyter notebook from your aws machine, tunneled to your computer. Please put in the token, if required. You will then be able to access the folder of your virtual machine.

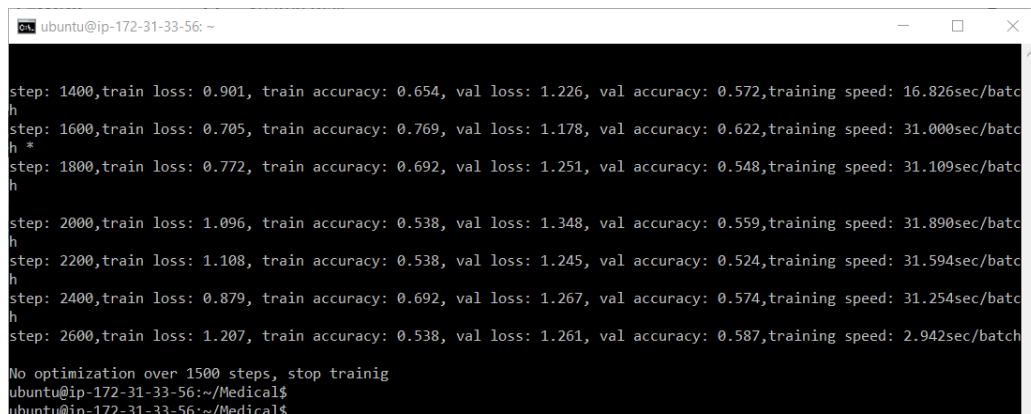


2. On your local computer, zip the main folder with your code, the pre-trained BERT model and the i2b2 corpus. Then, upload the folder into Jupyter notebook.
3. On your aws machine, unzip the file: if it is called Medical.zip, for example, use the command `unzip Medical.zip`. If it says you don't have the zip command, download it with `sudo apt-get install unzip`
4. You now have all the files you need in your aws machine to run the code.

Running the Code

Training:

1. To run the training code, in your aws machine, using the command line, go to the root directory where your downloaded code is, for example, `cd Medical`. Type in the command `dir` to make sure that files `data_process_i2b2.py` and `run_i2b2.py` are in that directory.
2. Create a new tmux session: this will allow you to access the code that's running even if you turn off your local computer or end the remote connection session. To do this, run the command: `tmux new -s mywindow`
3. Run the following command to process the data, it should only take a few minutes to finish: `python3 run_i2b2.py train`
4. Run the following command to train the data. It should take about 54 hours. In order to check the status, just connect to your aws machine through ssh and run the command `tmux a -t mywindow` and it will take you to the same session that you were in before. Please keep your aws machine connected and running.
5. Once it's finished, it will print in your session that the training has been completed.



```
ubuntu@ip-172-31-33-56: ~
step: 1400,train loss: 0.901, train accuracy: 0.654, val loss: 1.226, val accuracy: 0.572,training speed: 16.826sec/batch
step: 1600,train loss: 0.705, train accuracy: 0.769, val loss: 1.178, val accuracy: 0.622,training speed: 31.000sec/batch
step: 1800,train loss: 0.772, train accuracy: 0.692, val loss: 1.251, val accuracy: 0.548,training speed: 31.109sec/batch
step: 2000,train loss: 1.096, train accuracy: 0.538, val loss: 1.348, val accuracy: 0.559,training speed: 31.890sec/batch
step: 2200,train loss: 1.108, train accuracy: 0.538, val loss: 1.245, val accuracy: 0.524,training speed: 31.594sec/batch
step: 2400,train loss: 0.879, train accuracy: 0.692, val loss: 1.267, val accuracy: 0.574,training speed: 31.254sec/batch
step: 2600,train loss: 1.207, train accuracy: 0.538, val loss: 1.261, val accuracy: 0.587,training speed: 2.942sec/batch
No optimization over 1500 steps, stop training
ubuntu@ip-172-31-33-56:~/Medical$
ubuntu@ip-172-31-33-56:~/Medical$
```

6. If you will not immediately start testing, please stop your aws machine so you won't be charged until you use it again.

Testing:

1. In order to test the code, in your virtual machine, access your tmux session with `tmux a -t mywindow` if it is still running. If it's not, create a new one with `tmux new -s mywindow`.
2. Using the command line, go to the directory where your code is and run the following command: `python3 run_i2b2.py test`.

3. You will get some warnings, but, unless there is an error, it will continue testing. This could take a few hours.
4. Once you are able to write commands again, the testing should be finished.
5. Zip the results directory: inside the directory where your code is, run `zip -r result.zip result.` This will create a zipped folder with the results of your trained model.
6. If you want to download the trained model, run another Jupyter notebook with the command `jupyter notebook -port 8080` and access it on your computer like you did when you uploaded your data. You can then download the results file to your local computer.

