Practical Assignment 1 - Part-of-speech (PoS) Tagging

User Manual

Marcelo Ferrer

Aymen Merchaoui

To start using the application, the files of the assignment should be copied to a google drive folder. Once they are in the drive, the file with the extension .ipynb should be open with Google Colab.

The colab application is divided in 8 sections, one for the Imports and Declarations, then 6 for the different languages models (basic and advanced) and finally the conclusions. In figure 1 a simplified structure can be visualized without the advance models.

It's important to emphasize that all cells should be run in a sequential mode (especially inside each section). Only the models could be executed independently from the others but not from the imports and declarations section.

Same with conclusions, to execute this section, all previous 7 have to be already executed.

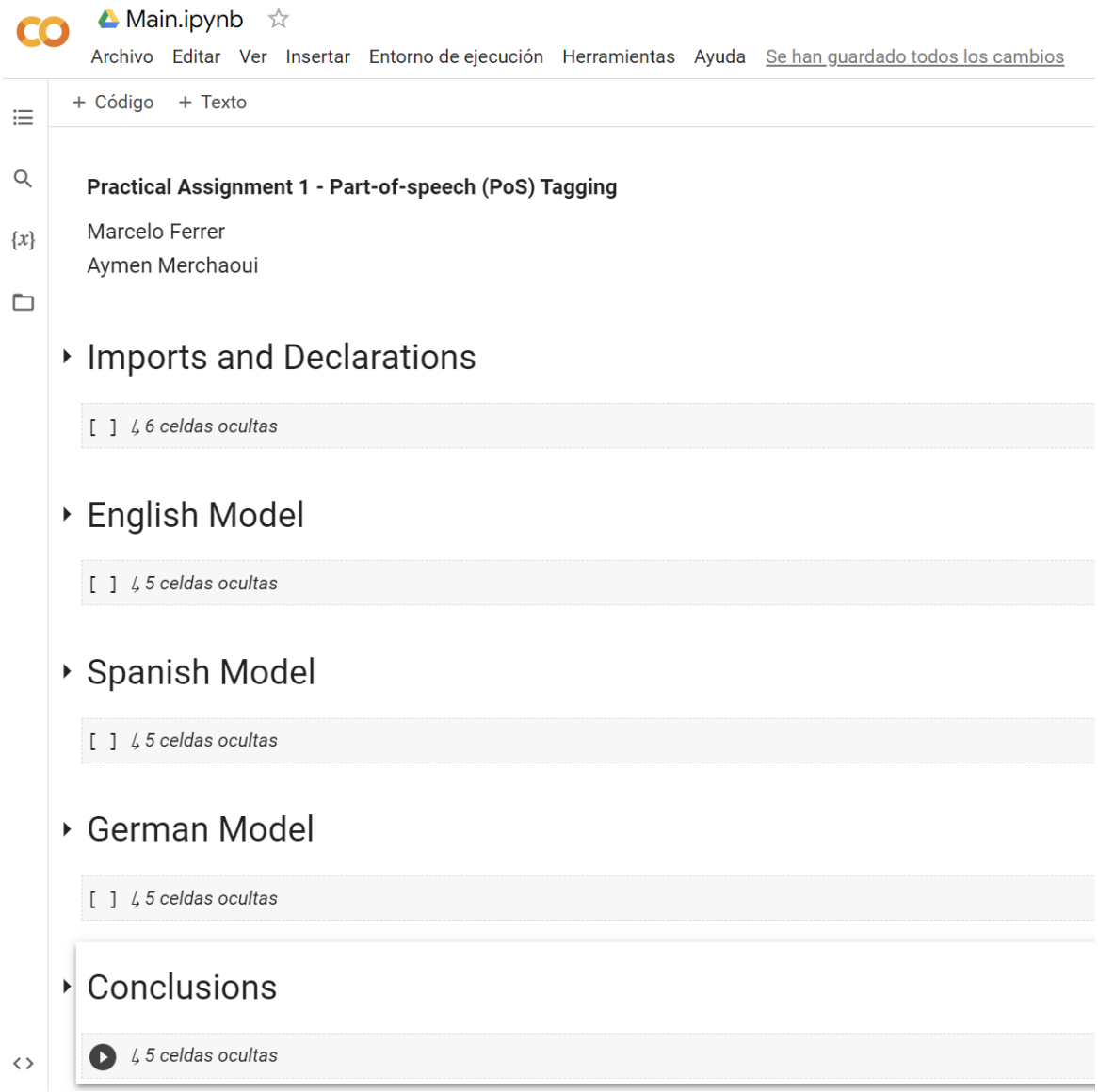

Figure 1 – Simplified Sections of the main

If the Imports and Declaration segment is expanded, we can see that the first cell installs conllu, a python library that parses Universal Dependencies files. The execution of this cell could take a few minutes.

Other libraries as Keras, Numpy, Matplotlib or Path, used in this assignment, should already be installed in the environment.

In the second cell (constants definitions), the first constant has the path to the folder where all the files are located (same folder as the one that the colab machine is located). This should be replaced with the path utilized by the user (Figure 2).

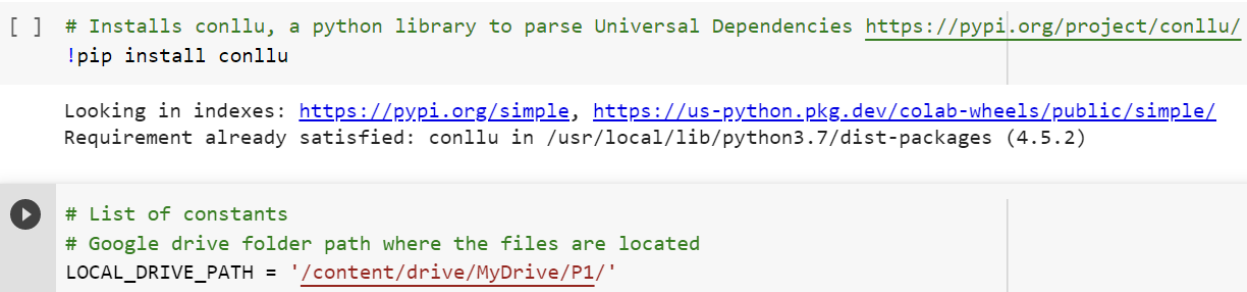

Figure 2 - Dependencies

Then, the third cell mounts the Google drive. When the execution of this cell is done, a popup asking about credentials to access may appear (Figure 3).

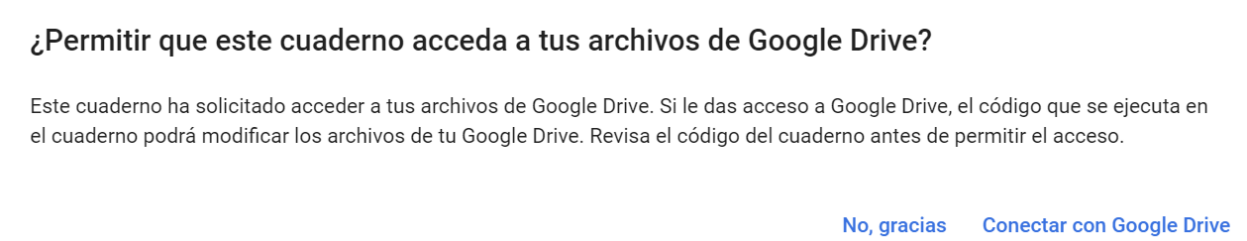

Figure 3 - Google drive access

The fourth cell execution (Figure 4) will grant access to the folder where the rest of the files are located which was defined in previous steps.

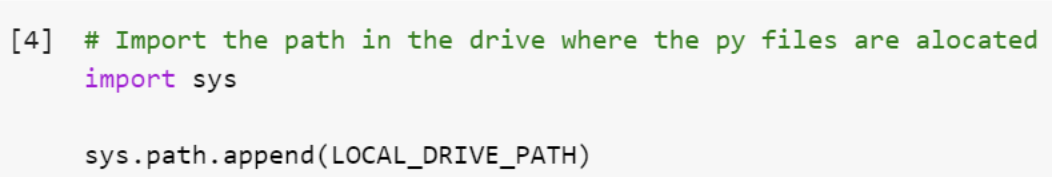

Figure 4 - File location

Once the drive is mounted and the local path appended, we can access the rest of the files from the menu at the left, pressing the files icon. A files tree will be deployed where in addition to the ipynb file, other files with the extension .py could be seen. (Figure 5)

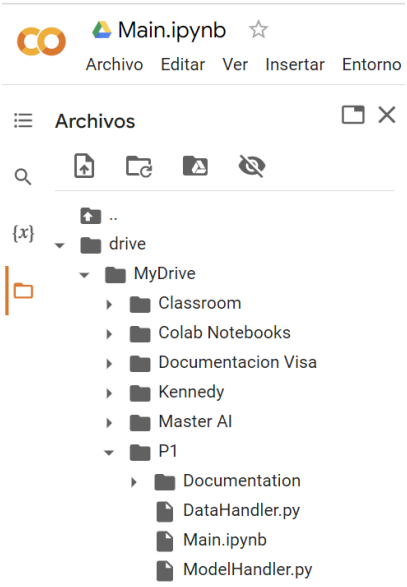

Figure 5 - File tree

In case the code of these files wants to be visualized or even changed, a simple double click on the file will open it in a new window on the right. (Figure 6)

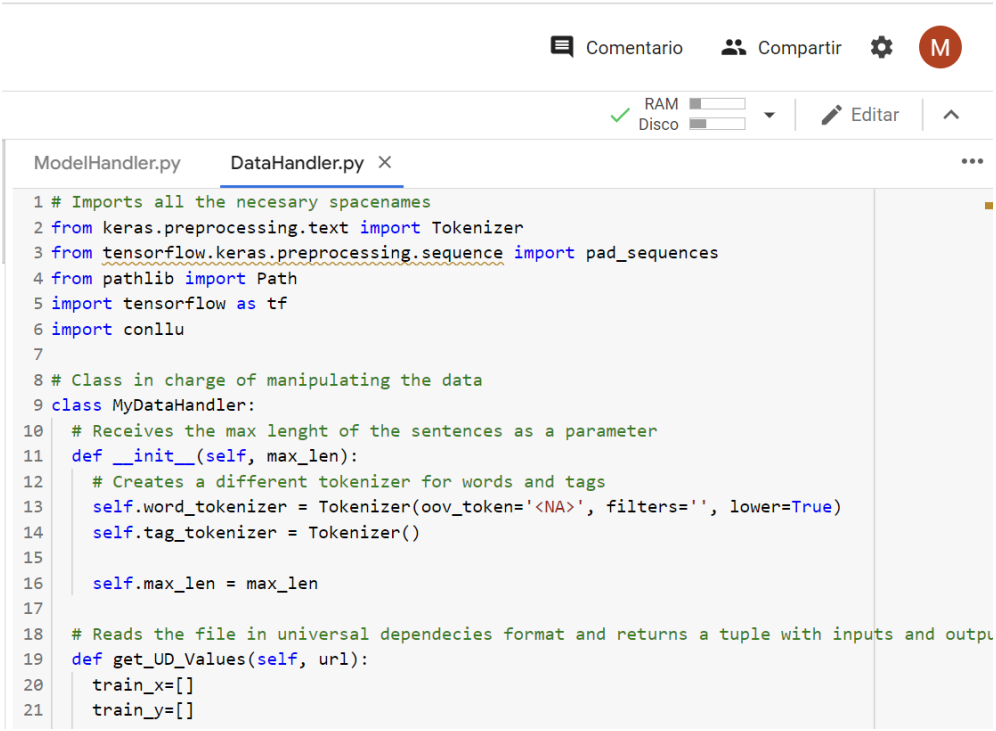

Figure 6 - Python files

In case a change has to be made to the python classes, it would be necessary to restart the environment so the main program can reload these differences. For that, in the top menu the option Execution Environment has a restart option. (Figure 7).

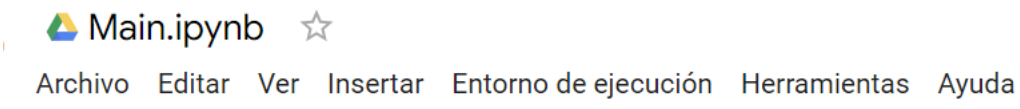

Figure 7 - Main menu

Another thing to take in account is that the training of the models can be a little slow, so having the GPU acceleration option of the environment activated could be a good idea.

For this, in the same bar of the Execution Environment there is a menu for changing the type of the execution environment. When selecting this option, a pop up with a combo box will be displayed, inside that, the GPU option is available. (Figure 8).

Note that this option is not limitless, so in case that more GPU power is needed, an option is to access with a premium account or opening the project with a different google user.

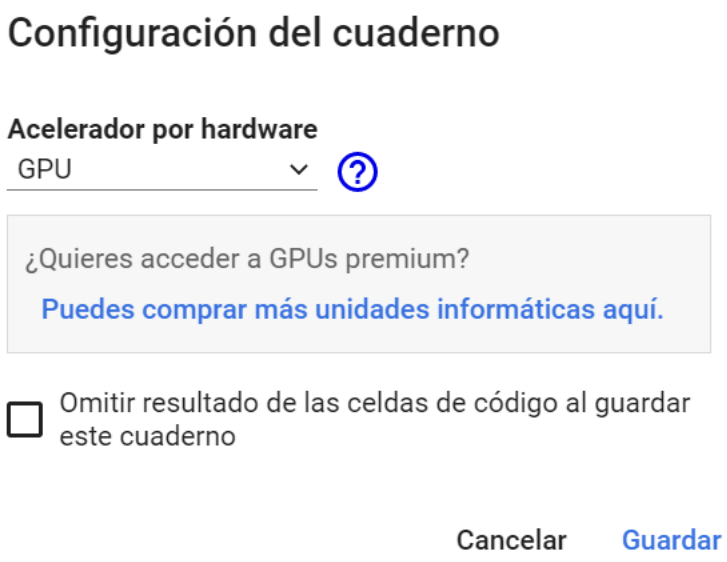

Figure 8 - Hardware accelerator

As stated before, each language model can be executed alone. The last cell of each one of these six sections has a User Input Sentences Function that allows the user to predict a vector of sentences with the trained model. (Figure 9).

User Input Sentences Function

```
[10] # To compute the part-of-speech tags for given new, previously unseen sentences inputted by the user.
     text_to_predict = ["Google is a nice search engine.", "Mary has a cat.", "Spiderman exists.", "I am real."]

     for x in text_to_predict:
       res = dh_eng.get_Detokenized_Values([mh_eng.test_predict(dh_eng.get_Test_Data(x))], "Output")
       print(x)
       print(res)
```

```
1/1 [==============================] - 0s 17ms/step
Google is a nice search engine.
['propn aux det adj noun noun punct']
1/1 [==============================] - 0s 17ms/step
Mary has a cat.
['propn verb det noun punct']
1/1 [==============================] - 0s 15ms/step
Spiderman exists.
['propn verb punct']
1/1 [==============================] - 0s 17ms/step
I am real.
['pron aux adj punct']
```

Figure 9 - User input sentences

Finally, in the section Imports and Declarations, other constants can be modified to try different parameters or languages. (Figure 10). An interesting constant is SHOW_SAMPLES, which allows the visualization of samples of the training, development and test sets and summaries of the different models.

The CURRENT_MODEL allows the selection of the different pre charged models in the ModelHandler, the USE_RNN allows replacing RNNs for LSTM layers and the DENSE_LEN creates an extra dense layer if it is greater than zero.

```
# Model to use in the execution
CURRENT_MODEL = "Functional_Bi"
# Use a RNN layer instead of a LSTM layer
USE_RNN = False
# Agregate an extra dense layer of N neurons
DENSE_LEN=128
# Optimizer to use in the compilation of the model
OPTIMIZER = "RMSProp"
# Maximun length of the sequences
MAX_LEN = 128
# Size of the batch to use in the data training
TRAIN_DATA_BATCH_SIZE = 128
# Embedding size
EMBEDDING_SIZE = 128
# Size of the batch to use in the training
BATCH_SIZE = 16
# Amount of epochs to train
EPOCHS = 10
# Print samples of the data and model
SHOW_SAMPLES = False
# Urls of the languagues
URL_TRAIN_ENG='https://raw.githubusercontent.com/UniversalDependencies/UD_English-EWT/master/en_ewt-ud-train.conllu'
URL_TEST_ENG='https://raw.githubusercontent.com/UniversalDependencies/UD_English-EWT/master/en_ewt-ud-test.conllu'
URL_VAL_ENG='https://raw.githubusercontent.com/UniversalDependencies/UD_English-EWT/master/en_ewt-ud-dev.conllu'
URL_TRAIN_SPA='https://raw.githubusercontent.com/UniversalDependencies/UD_Spanish-GSD/master/es_gsd-ud-train.conllu'
URL_TEST_SPA='https://raw.githubusercontent.com/UniversalDependencies/UD_Spanish-GSD/master/es_gsd-ud-test.conllu'
URL_VAL_SPA='https://raw.githubusercontent.com/UniversalDependencies/UD_Spanish-GSD/master/es_gsd-ud-dev.conllu'
URL_TRAIN_GER='https://raw.githubusercontent.com/UniversalDependencies/UD_German-GSD/master/de_gsd-ud-train.conllu'
URL_TEST_GER='https://raw.githubusercontent.com/UniversalDependencies/UD_German-GSD/master/de_gsd-ud-test.conllu'
URL_VAL_GER='https://raw.githubusercontent.com/UniversalDependencies/UD_German-GSD/master/de_gsd-ud-dev.conllu'
```

Figure 10 - Constants