

Assignment-0

Artificial Intelligence (CSO-241)
IIT (BHU) Varanasi

January 9, 2020

1 Create Python virtual environment

Follow the steps given in the link to create a Python (version 3.5+) virtual environment.

Link : [Tutorial](#)

2 Installing libraries

2.1 Python libraries

You have to use the virtual environment created in the first step for downloading libraries. With this assignment you are provided **requirements.txt** file. Type the following command in your virtual environment.

```
pip install -r requirements.txt
```

2.2 CRF++

2.2.1 For Windows

Download CRF++ 0.58 from this [link](#)

2.2.2 For Linux/MacOS

Download CRF++ 0.58 from this [link](#)

For installation follow the steps below:

Extract the tar.gz archive and run the following commands:

```
cd your_path_here/CRF++-0.58
```

```
./configure
```

```
make
```

```
sudo make install
```

Alternatively follow this [link](#)

2.3 Cascade Trainer GUI

This is a GUI tool that can be used to train, test and improve cascade classifier models. This tool is only available for Windows.

2.3.1 Instructions to download

Follow this link to the official download page [link](#)

3 Weka

3.1 For Windows

Click [here](#) to download. This will install Weka executable in your Program Menu.

3.2 For Linux

Click [here](#) to download a zip archive for Linux

3.3 For MacOS

Click [here](#) to download a disk image for Mac OS

4 Verifying the installation

4.1 Python libraries

To check Python libraries correctly installed, run the python script **check_installation.py** given with this assignment.

4.2 CRF++

4.2.1 For Windows

Just ensure that you have **crf_learn.exe** and **crf_test.exe** files in the extracted folder.

4.2.2 For Linux/MacOS

Type the following commands:

1. `./crf_learn`

You should get output like this:

```
ankan@ankan-HP-Notebook: ~/Desktop/AI/CRF++-0.58
ankan@ankan-HP-Notebook:~/Desktop/AI/CRF++-0.58$ ./crf_learn
CRF++: Yet Another CRF Tool Kit
Copyright (C) 2005-2013 Taku Kudo, All rights reserved.

Usage: /home/ankan/Desktop/AI/CRF++-0.58/.libs/crf_learn [options] files
-f, --freq=INT          use features that occur no less than INT(default 1)
)
-m, --maxiter=INT       set INT for max iterations in LBFGS routine(default
10k)
-c, --cost=FLOAT        set FLOAT for cost parameter(default 1.0)
-e, --eta=FLOAT         set FLOAT for termination criterion(default 0.0001)
-C, --convert           convert text model to binary model
-t, --textmodel         build also text model file for debugging
-a, --algorithm=(CRF|MIRA) select training algorithm
-p, --thread=INT        number of threads (default auto-detect)
-H, --shrinking-size=INT set INT for number of iterations variable needs to
be optimal before considered for shrinking. (default 20)
-v, --version           show the version and exit
-h, --help             show this help and exit

ankan@ankan-HP-Notebook:~/Desktop/AI/CRF++-0.58$
```

Figure 1: Testing `crf_learn`

2. `./crf_test`

You should get output like this:

```
ankan@ankan-HP-Notebook: ~/Desktop/AI/CRF++-0.58
ankan@ankan-HP-Notebook:~/Desktop/AI/CRF++-0.58$ ./crf_test
feature_index.cpp(193) [mmap_.open(model_filename)] mmap.h(153) [(fd = ::open(fi
lename, flag | O_BINARY)) >= 0] open failed:
ankan@ankan-HP-Notebook:~/Desktop/AI/CRF++-0.58$
```

Figure 2: Testing `crf_test`

4.3 Weka

4.3.1 For Linux

To run Weka, change into that directory and type

```
./weka.sh
```

You should get output like this:

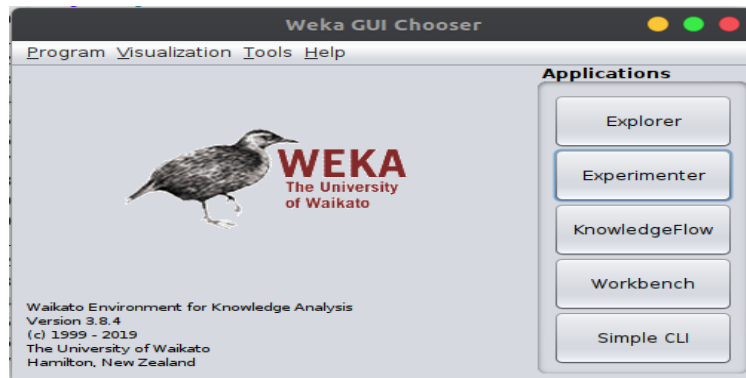


Figure 3: Weka GUI

4.3.2 For Windows

Run the exe file after installation.

4.4 Cascade Trainer GUI

4.4.1 For Windows

You should see the following window after installation.

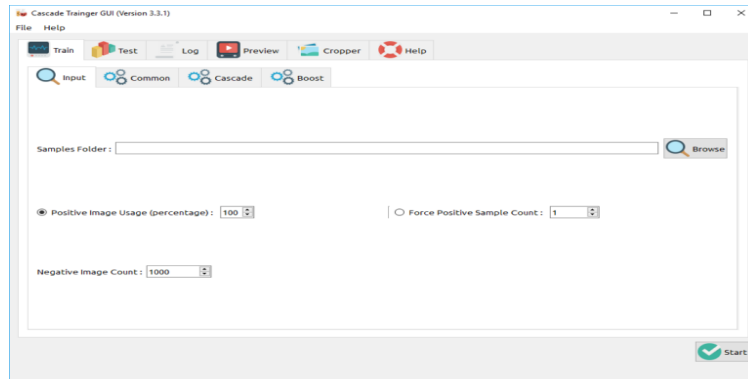


Figure 4: Cascade Trainer GUI

NOTE : If you have any queries, ask on the Google Classroom.