CDC Services Installation Instructions

This documentation contains the installation instructions for all the services HLA has provided for CDC Project. In general, there are 4 types of services we provide:

* HLAPostProcessors for LAPPS
* CER Pipelines
* CER Server
* Feature Library Web Page

# HLAPostProcessors for LAPPS

Python Packages Required:

* lxml
* requests

All the source codes are in the CDC Gitlab:

<https://git.phiresearchlab.org/dongjen/HLALAPPSTools>

Each module to be installed into LAPPS contains one python file and one xml configuration file. There are 8 post processors to be installed in total.

First, you need to copy all the source codes in HLAPostProcessors into galaxy/mods/tools directory where all the files for LAPPS tools are stored.

# If you are in the home directory

cd galaxy/mods

cp –r /home/user/HLAPostProcessors tools

Second, the tool configuration in the file galaxy/mods/config/tool\_conf.xml needs to be extended with the HLAPostProcessors configuration. The below new section configuration needs to be added to the configuration file.

<section id=”postprocessors” name=”HLAPostProcessors”>

<tool file=”HLAPostProcessors/PostTokenizer\_GATEDef.xml”/>

<tool file=”HLAPostProcessors/PostTokenizer\_StanfordDef.xml”/>

<tool file=”HLAPostProcessors/PostTokenizer\_OpenNLPDef.xml”/>

<tool file=”HLAPostProcessors/PostSentenceSplitter\_GATEDef.xml”/>

<tool file=”HLAPostProcessors/PostSentenceSplitter\_StanfordDef.xml”/>

<tool file=”HLAPostProcessors/PostSentenceSplitter\_OpenNLPDef.xml”/>

<tool file=”HLAPostProcessors/POSPostProcessDef.xml”/>

<tool file=”HLAPostProcessors/FeatureExtractor\_Def.xml”/>

<tool file=”HLAPostProcessors/FeatureExtractor\_GATEDef.xml”/>

</section>

After all these installation steps are finished, you need to restart LAPPS for these new tools to work. An example workflow of how these post processors work has been shared in LAPPS. The command to run them locally is described in the README.md.

# CER Pipelines

Python Packages Required:

* lxml
* requests
* zeep
* subprocess
* corenlp
* nltk

All the source code is on CDC Gitlab:

<https://git.phiresearchlab.org/dongjen/HLACERPipeline>

The main installation process for CER Pipelines can be separated into three parts:

1. Install appropriate Python and packages stated above
2. Install CRF
3. Install StanfordNLP Packages

## Install appropriate Python and packages stated above

The CER Pipelines have been tested to run successfully on Python 3.5 +. The following command can be used to install these Python packages.

pip3 install lxml

pip3 install requests

pip3 install zeep

pip3 install subprocess

pip3 install nltk

## Install CRF

The source files of CRF have been put under the Training folder. The installation can be done by running the command below inside the CRF++-0.58 directory. The requirement of this installation is to have a C++ compiler (gcc 3.0 or higher).

./configure

make

sudo make install

## Install StanfordNLP Packages

Firstly, the Stanford CoreNLP can be downloaded and unzipped from the official releases using the following command:

wget <http://nlp.stanford.edu/software/stanford-corenlp-full-2016-10-31.zip>

unzip stanford-corenlp-full-2016-10-31.zip

You might need to install unzip if you haven’t already done so.

sudo apt install unzip

Secondly, you need to define an environment variable $CORENLP\_HOME that points to the unzipped directory. The Python Stanford CoreNLP package can then be installed using the following command:

pip3 install stanford-corenlp

After the installation is completed, the CER Pipelines can be run using the following command.

nohup python3 –u Stanford.py > stanford.out &

nohup python3 –u OpenNLP.py > opennlp.out &

nohup python3 –u GATE.py > gate.out &

nohup python3 –u cTAKE.py > ctake.out &

A more detailed description of the input and output format is written in the README.md of the Git repository.

# CER Server

The CER Server takes the text of the report as input, uses the generated model to tag the file and generates a LIF format file with all the tagged instances.

Python packages required:

* lxml
* requests
* zeep
* bottle
* threading

All the source code is on CDC Gitlab:

<https://git.phiresearchlab.org/dongjen/HLACERServer>

The installation process of CER Pipeline needs to be completed before installing the CER Server. In order to allow external access to the server, CER Server needs to be installed using Apache2.

Apache2 needs to be installed in order to install the server. There are 3 servers to be setup, Stanford Server, OpenNLP Server and GATE Server.

If the /etc/apache2/conf-available/httpd.conf configuration file does not currently exist, you need to create the file and add the following line to the apache2 configuration /etc/apache2/apache2.conf.

IncludeOptional conf-enabled/\*.conf

The following lines need to be added to the /etc/apache2/conf-available/httpd.conf file. You will need to change the path to the wsgi file accordingly.

# Server Address: http://hostname.com/cdc\_service\_stanford

WSGIDaemonProcess cdc\_service home=/home/jenny/CDC\_Server

WSGIScriptAlias /cdc\_service\_stanford /home/jenny/CDC\_Server /stanford\_server.wsgi

<Directory /home/jenny/CDC\_Server>

WSGIProcessGroup cdc\_service

WSGIApplicationGroup %{GLOBAL}

Require all granted

</Directory>

# Server Address: http://hostname.com/cdc\_service\_opennlp

WSGIDaemonProcess cdc\_service\_opennlp home=/home/jenny/CDC\_Server

WSGIScriptAlias /cdc\_service\_opennlp /home/jenny/CDC\_Server /opennlp\_server.wsgi

<Directory /home/jenny/CDC\_Server>

WSGIProcessGroup cdc\_service\_opennlp

WSGIApplicationGroup %{GLOBAL}

Require all granted

</Directory>

# Server Address: http://hostname.com/cdc\_service\_gate

WSGIDaemonProcess cdc\_service\_gate home=/home/jenny/CDC\_Server

WSGIScriptAlias /cdc\_service\_gate /home/jenny/CDC\_Server /gate\_server.wsgi

<Directory /home/jenny/CDC\_Server>

WSGIProcessGroup cdc\_service\_gate

WSGIApplicationGroup %{GLOBAL}

Require all granted

</Directory>

After that, the Apache2 needs to be restarted for the above configuration to be in effect. The following command can be used to restart Apache2.

sudo service apache2 restart

After the installation is finished, the server can be called using the script provided in CDC\_Server\_Client directory.

# Feature Library Web Page

The Feature Library web page takes the folder of text or ann files and generates the resulting BIO files with all the features users selected.

Python packages required:

* lxml
* requests
* zeep
* bottle
* corenlp
* nltk
* zipfile
* pydot

All the source code is on CDC Gitlab:

<https://git.phiresearchlab.org/dongjen/HLAFeatureLibrary>

Before installing the Feature Library, the CER Pipeline needs to be installed. The path to bin of the graphvix-2.38 needs to be added into the PATH environment variable. In Python, this can be done in the following command. The path needs to be changed accordingly.

os.environ["PATH"] += os.pathsep + '/home/jenny/CLEW\_Feature\_Library/graphviz-2.38/release/bin'

After that, the Feature Library needs to be installed as two parts in Apache2: back-end server and front-end web page.

The following configuration settings need to be added to the /etc/apache2/conf-available/httpd.conf configuration file. You will need to change the path to the wsgi file accordingly.

# Configure back-end server

# Server Address: http://hostname.com/clew\_feature\_library

WSGIDaemonProcess feature\_library home=/home/jenny/CLEW\_Feature\_Library

WSGIScriptAlias /clew\_feature\_library /home/jenny/CLEW\_Feature\_Library/feature\_library\_service.wsgi

<Directory /home/jenny/CLEW\_Feature\_Library>

WSGIProcessGroup feature\_library

WSGIApplicationGroup %{GLOBAL}

Require all granted

</Directory>

After installing the back-end server, its url needs to be changed in the Demo\FeatureLibrary\_Caller.py line 18 and line 45.

# Configure front-end web page

# http://hostname.com/clew/feature\_library

WSGIDaemonProcess feature\_library\_demo home=/home/jenny/CLEW\_Feature\_Library/Demo

WSGIScriptAlias /clew /home/jenny/CLEW\_Feature\_Library/Demo/Feature\_Library\_Demo.wsgi

<Directory /home/jenny/CLEW\_Feature\_Library/Demo>

WSGIProcessGroup feature\_library\_demo

WSGIApplicationGroup %{GLOBAL}

Require all granted

</Directory>

After the installation is completed, the feature library web page can be accessed in the url <http://hostname.com/clew/feature_library/main>

The screenshot of final feature library web page is presented below.



