Guideline for Training ParsCit model for Reference Parsing.

Steps for training Reference Parsing model:

This documentation elaborates only on the Reference Parsing model and not on the Reference Extraction. This is because ParsCit uses a 3rd party tool ***Omnipage*** for converting the PDF to its appropriate XML format.

Refer to following link for the detailed documentation of ParsCit model. [1]http://parscit.comp.nus.edu.sg/  
  
Depending on the type of the operating system download the appropriate version of the model as mentioned in the link above.

ParsCit uses different Perl libraries in background for its functioning. The complete list of libraries used can be found in the INSTALL.md file. Please install all of them.

ParsCit uses CRF learner for the reference parsing model. It is required to build the CRF++ library before using the model for training. The instructions for building the CRF++ library are as follows:

* $ cd crfpp
* $ rm -Rf CRF++-0.51
* $ tar -xvzf crf++-0.51.tar.gz
* $ cd CRF++-0.51
* $ ./configure
* $ make
* $ make clean
* $ make
* $ sudo make install
* Please note that during this process you might face some difficulties in compiling the libraries.
* Apparently, the error message is not that intuitive. However, you can add following suggestions to compile the CRF++ libraries.  
  add the following import statements before the beginning of the import list in ***node.cpp*** & ***path.cpp*** files.  
  **#import <iostream.h>  
  #import stdlib.h**
* Hopefully by adding the above-mentioned import statements ***crf\_test & crf\_learn*** files are created.
* move executables to where parscit expects to find them
* $ cp crf\_learn crf\_test ..
* # on Windows you may have to do this instead, as the executables are named with .exe
* $ copy crf\_learn.exe ../crf\_learn
* $ copy crf\_test.exe ../crf\_test
* $ cd .libs
* $ cp -Rf \* ../../.libs

Once the libraries are placed properly, if you are not using the CRFPP\_HOME environment variable, you may need to edit the lib/ParsCit/Config.pm file to point to the proper directories on your machine. If you are using CRFPP\_HOME, you may want to set it in your .bashrc or .profile file.

Try running the ***parseRefStrings.pl*** file on the .txt data found in demo-data using the following parameters.

**$ perl -CSD parseRefString.pl inputfile.txt > outputfile.txt**

(parseRefString.pl -> takes 1 argument the input file containing the raw reference strings and if the output of the process needs to be stored to a file then specify the name of the file as the 2nd argument i.e. outputfile.txt in the above command.

Additionally, if k-fold cross validation needs to be performed then a simple ***shell*** script can be written to perform the training and evaluate steps.

Preparing data for the training process.

ParsCit expects the training data to be formatted in a specific format that is understood by the model. The appropriate training data can be generated by using the templates provided by parscit. The following steps elaborates on the steps required to do so.

1. Parscit does not perform granular segmentation of Bibliographic element author. I.e. surname and given-names/ first-name of the author are not considered. Instead a single ***<author>*** tag is used to represent the authors found in the reference string.
2. Similarly, ***<pages>*** tag is used to represent the Bibliographic element ***<first-page> & <last-page>*** and ***<article-title> tag*** is replaced by ***<title> tag.***
3. All tag must be formatted in this format  
   for e.g. : <tag>space ***token*** </tag>

<author> Abbink J., </author> <year> 2006 </year> <title> Discomfiture of democracy? The 2005 election crisis in Ethiopia and its aftermath </title> <journal> African Affairs </journal> <volume> 105 </volume> <pages> 173 - 199 </pages>  
**If this format is not followed then the appropriate data file will contain training features which will not be well parsed by the CRF ++ classifier resulting in training error**.

\*\*\* Also note that the training data is to be placed under following directory structure.

**crfpp/traindata/**

1. Once the training data is prepared in the space separated format as specified in step 3, we must obtain the training features for the CRF++ classifier. The following set of commands will achieve the task of a) feature extraction , b) model training c) placing the trained model in the appropriate directory   
    **a) $ ../../bin/tr2crfpp.pl tagged\_references.txt > parsCit.train.data**tagged\_references.txt : is the file containing the references  
   parsCit.train.data : the data file containing the training features

**b) $ ../crf\_learn parsCit.template parsCit.train.data model**parsCit.template : it is the template file for CRF++

parsCit.train.data : the data file generated in step a.  
model : CRF++ trained model. Note that the name of the model is “model” which can be replaced by just specifying the name of choice.

**c) $ mv model ../../resources/parsCit.model**This command will replace the “model” generated in step b) with resources/parsCit.model.