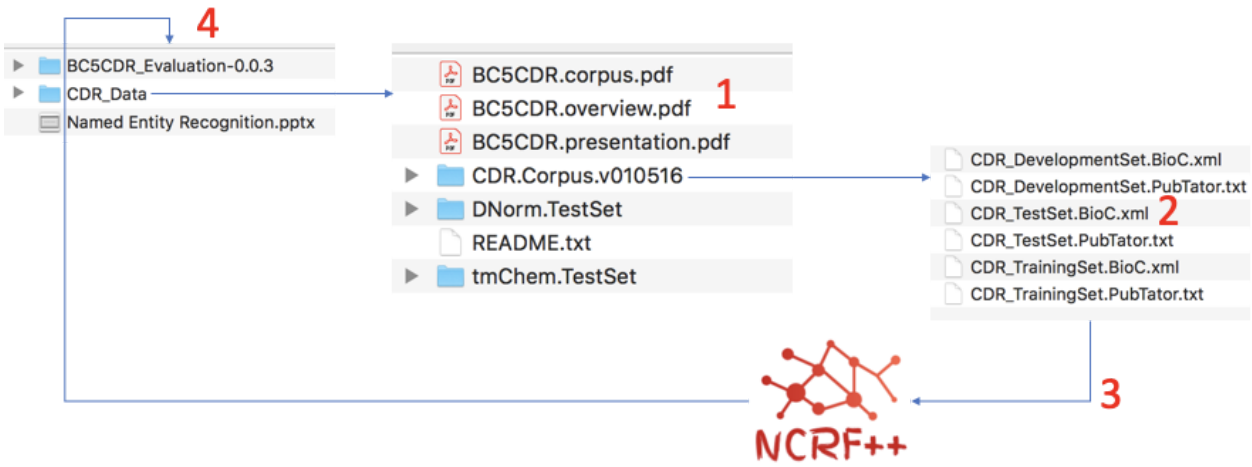
Digital Health Project 2

Shaozhi Jiang
Student ID: 01775431
April 30, 2019

1. Goal:

This project is to build a module to inference diseases.
Input: CDR_Data
Output: inference result of CDR_TestSet

- Data: BioCreative V Chemical-Disease Relation (CDR) Task Corpus



2. Solutions

I use PubTator Format to train model, and inference. There are 5 steps, including pre-process data, train module, decode data, generate result with module and evaluate.

Environment:

Python: 2.7
Pytorch: 1.0.1
OSX : 10.1
NCRF++

1) Preprocess CDR data
   Preprocessed CDR data to specific data format, via "pre-process.py.py"
   from: "CDR_DevelopmentSet.PubTator.txt",
   "CDR_TrainingSet.PubTator.txt","CDR_TestSet.PubTator.txt"
   to: "dev.txt","train.txt","test.txt"
   and test_info.txt

2) Train Module in NCRF++
Configed NCRF++ to train without GPU,
Configure file: "dh.train.config"
Output module file: "lstmcrf.6.model"

3) Decode data in NCRF++
Configed NCRF++ to train without GPU,
Configure file: "dh.decode.config"
Output file: "raw.out"

4) Generate result
generate result via "assemble.py"

5) Evaluation
Evaluated via bc5cdr_eval.jar

Here are the run script.

```
echo " pre-process CDR data, split to datasets: dev.txt, train.txt, test.txt "
echo " done. python NERPrj/pre-process.py.py "
cd NERPrj
python pre-process.py.py

echo "--------------"
echo "start training.. to get model "
cd ..
python main.py --config dh.train.config


echo "--------------"
echo "start decode... to  get rawout "
python main.py --config dh.decode.config

echo "--------------"
echo "process rawout  to get result.txt"
python NERPrj/assemble.py


echo "--------------"
echo " evaluate the result"
cd NERPrj
./eval_mention.sh PubTator inputData/CDR_TestSet.PubTator.txt outputData/result.txt
cd..
```

3. Result:

Train: (100 epoches)

```
     Instance: 6521; Time: 19.40s; loss: 1.1413; acc: 114543.0/114572.0=0.9997
Epoch: 99 training finished. Time: 51.65s, speed: 126.26st/s,  total loss: 3.07726867675
totalloss: 3.07726867675
Right token =  107100  All token =  113768  acc =  0.94138949441
Dev: time: 12.19s, speed: 557.64st/s; acc: 0.9414, p: 0.7490, r: 0.6580, f: 0.7006
Right token =  112941  All token =  119789  acc =  0.942832814365
Test: time: 13.20s, speed: 561.91st/s; acc: 0.9428, p: 0.7438, r: 0.6595, f: 0.6991
```

Decode:

```
Decode raw data, nbest: None ...
Right token =  125561  All token =  130576  acc =  0.961593248376
raw: time:20.13s, speed:368.20st/s; acc: 0.9616, p: 0.7765, r: 0.6932, f: 0.7325
Predict raw result has been written into file. NERPrj/outputData/raw.out
(DigitalHealth_PY2.7) MacBook-Pro:NCRFpp michaeljiang$
```

Evaluation:

```
(DigitalHealth_PY2.7) MacBook-Pro:NERPrj michaeljiang$ ./eval_mention.sh PubTator inputData/CDR_TestSet.PubTator.txt outputData/result0.txt
TP: 164
FP: 3831
FN: 4260
Precision: 0.04105131414267835
Recall: 0.037070524412296565
F-score: 0.03895949637724195
```

The steps of preprocess, train, decode are pretty well, but evaluation are not that good because there are some issue of positions in punctuations.

4. Files:

info_test.txt	result.txt	
assemble.py	info_train.txt	runme.sh
dev.txt	lstmcrf.6.model	test.txt
dh.decode.config	lstmcrf.dset	train.txt
dh.train.config	pre-process.py	~$ite-up.docx
info_dev.txt	raw.out	