Supervised Clinical Document Classification Pipeline

Prerequisite

- Linux or Mac environment, Python (≥ 2.6) Perl (≥ version 5) for deidentification
- For concept identification and parsing using cTAKES - Java (≥ 1.7), cTAKES and its UMLS dictionary, register UTS service for cTAKES concept identification, connect to the Internet

Quick Start

1. Clone the github repository → `sudo python setup.py install` → go to subdirectory `/cdc/bin/` and run `sh test_model.sh`. The testing script will use sample iDASH dataset, which will be automatically downloaded.
2. Modeling (modifying `test_model.sh`)
   - `python ../src/modeling.py -w  /test/ -c xml -m T -f bow+sg+bow_sg -v freq -a l1+l2+nb+svmlin+svmrbf+rf+adaboost+gbm -b None -r 3 -k 5`
     - The working directory (`-w`) is `/test/`, without data source (`-d`) (so the pipeline will download the sample files). The sample iDASH files are in XML format so we used `xml` in `-c`. Here we use frequency count for vector representation (`-v`), and check the performances of all algorithms (`-a`).
   - `python ../src/modeling.py -w  /test/ -d  /data/ -c xml -l  /label.txt -p cTAKES -q  /CTAKES_HOME/ -s T -n NAME -o PWD -e 2g -g 6g -m T -f sg+st -v freq -a l1 -r 3 -k 5`
     - The data directory is `/data/`. We want to use semantic concepts (`-f` is `sg+st`), so `-p` and `-q` arguments should be required, and the `-c` argument should be `xml`. `-s, -n, -o, -e, -g` are used for first time cTAKES setting. In this case we run regularized logistic regression with 3 repeated 5-fold cross-validation.
   - `python ../src/modeling.py -w  /test/ -d  /data/ -c txt -l  /label.txt -m T -f bow -v freq -a l1 -b None -r 1 -k 5`
     - The data directory is `/data/`. In this case we just want to use bag-of-words (`-f` is `bow`) instead of concept parsing (other than `bow`), so `-p` and `-q` arguments are not required, and the `-c` argument should be `txt` for `bow`. Without concept parsing, only `bow` can be assigned to the `-f` (feature) argument.
3. Output format
   - Three subfolders `data`, `model` and `result` will be created inside experiment working directory (you need to pass the path to `-w`. Model performance will be saved in `result` subfolder. The model with the best performance will be saved in the `model` subfolder in `.pkl` format. Feature and encoder will also be saved in the `model` subfolder.
4. Prediction (modifying `test_predict.sh`)
   - For prediction, we just need to assign the directory with your new data, and the path to the model created in the modeling step. It is usually under the subfolder `model` inside your working directory.
   - `python ../src/predict.py -d [YOUR DATA] -m [MODEL PATH (.pkl)]`
   - E.g. `python predict.py -d /Users/weng/Desktop/tt/ -m /Users/weng/idash/model/model_f=bow_a=LogisticR.pkl`
   - Need to add `-p ctake -q  /CTAKES_PATH/` if the model is not bag-of-words