Date de création : Vendredi 04 Mars 2016
Date de modification : Samedi 05 Mars 2016
Titre : Rapport d'installation Moses + accessoires
Auteur : Ngoc Tan LE

VirtualBox VM version 4.3.24
    Ubuntu 14.04 LTS (64 bits)
        Ram : 9 Go
        i5, 2.67 GHz
        266.60 Go Disque dur

Question :: How to install Moses on Ubuntu 64 bits ?
Sources ::
    http://www.achchuthan.org/2014/06/install-moses-on-ubuntu-14.04.html
    http://www.statmt.org/moses/?n=Moses.Baseline
    http://www.statmt.org/moses/?n=Development.GetStarted

5 STEPS ::

Step 1: Installing the following package using the commands

```
        *** ho tro cho ubuntu
        sudo apt-get install build-essential git-core pkg-config automake libtool wget zlib1g-dev g++ git subversion libboost-all-dev libbz2-dev liblzma-dev python-dev graphviz imagemagick make cmake  libgoogle-perftools-dev  libsoap-lite-perl
        libtcmalloc-minimal4
```

Step 2: Installing Boost

```
        // boost_1_60
        // j5 #core i5
        24  tar zxvf boost_1_60_0.tar.gz
        25  cd boost_1_60_0/
        28  ./bootstrap.sh
        29  ./b2 -j5 --prefix=$PWD --libdir=$PWD/lib64 --layout=tagged link=static threading=multi,single install || echo FAILURE

        #
        #
```

cu Tan cai dat moses chua dung. phai cai dat srilm 1.6.0, ko can dung irstlm, truoc khi cai dat moses. neu da cai dat moses roi thi van co the recompile.

+ neu cu e cai boost manually thi luc compile moses phai chi duong dan den boost. Tuy nhien, boost moi ko dc moses ho tro tot. Neu ku e ko chi path den boost da cai dat thi moses se dung boost co san trong source moses.

```
                #
                #

        Step 3: Installing MOSES :: mosesdecoder
                git clone https://github.com/moses-smt/mosesdecoder.git
                47  ls -l
                48  mv mosesdecoder-master mosesdecoder   # change the name of the folder if downloading directly via the Website
                51  cd mosesdecoder/
                52  ls -l

                ### Source Installation from statmt.org
                make -f contrib/Makefiles/install-dependencies.gmake



                53  ./bjam -j5    # by default hoac la ./bjam --with-boost=~/boost_1_60_0 -j5

                ./bjam  --with-boost=/home/tan/boost_1_60_0 -j5
                                --with-srilm=/home/tan/srilm -j5
                                --with-giza=/home/tan/mgiza -j5

        Step 4: Installing GIZA++ || MGIZA (chu y khi goi mgiza trong test Moses)
                // train-model.perl -mgiza -cpus <NUMBER>   # to specify the number of CPUs
                // train-model.perl -mgiza
                tai ve tu mgiza
                *** mgiza - Khong can cai dat GIZA
                cd mgizapp/
                sudo apt-get install cmake
                cmake .
                #sudo apt-get install libboost-all-dev
                #make -j4
                #cmake . -DCMAKE_INSTALL_PREFIX=/home/lent/Develops/Solution/tool/GIZA++
                make install

                cd ~/mosesdecoder
                mkdir tools
                cd tools
                cp ~/mgiza/mgizapp/scripts/merge_alignment.py  .   # mkcls, snt2cooc, merge_alignment.py   TRONG THU MUC TOOLS
                cp ~/mgiza/mgizapp/bin/* .
```

Step 5: Installing IRSTLM || SRIMLM || KenLM (par défaut)

```
tar zxvf irstlm-5-80-03.tgz   # dernière version : 5-80-08

cd irstlm-5-80-03
./regenerate-makefiles.sh
./configure --prefix=/home/tan/irstlm-5-80-03
make install

###
# INSTALL SRILM
        tai tren Website ve :: current version = 1.7.1
        unzip vao trong thu muc srilm
        trong tap tin Makefile, uncomment dong thu 7 :
                SRILM = /home/tan/srilm                 # chi duong dan absolute toi thu muc
        prompt >>> cd srilm
        prompt >>> make World                                   # cai dat SRILM, output :: thu muc bin duoc tao ra. Trong do co thu muc i686-m64 (he dieu hanh 64 bits, Linux).
        trong thu muc i686-m64 :
                ngram, ngram-class, ngram-count
###
```

Question :: How to test Moses on Ubuntu 64 bits ?
Sources ::
http://www.statmt.org/moses/?n=Moses.Baseline

http://www.statmt.org/moses/?n=FactoredTraining.HomePage

Training process

The nine steps are :

1. Prepare data : 45 min
2. Run GIZA++ : 16 hours
3. Align words : 2h30
4. Get lexical translation table : 30 min
5. Extract phrases : 10 min
6. Score phrases : 1h15
7. Build lexicalized reordering model : 1h
8. Build genereation models
9. Create configuration file : 1 sec

Corpus preparation ::
Source:

wmt13/training-parallel-nc-v8.tgz

fr-en : 157.168 sentences pairs

=== PRELIMINAIRE ===

tao thu muc :

- working chua cac tap tin training, test, evaluation of BLEU, NIST, TER, etc.
- corpus chua cac tap tin ngu lieu song ngu : raw, tokenised, truecase, clean
- mosesdecoder
- mgiza
- irstlm
- boost chua version boost_1_60_0 (the newest version) for Moses

====================

- tokenisation
- truecase
- cleaning : cutoff 1-80 ratio=9

=== CAU LENH ===

```
71  mosesdecoder/scripts/tokenizer/tokenizer.perl -l en < corpus/FrEn/training.en > corpus/FrEn/training.tok.en
72  mosesdecoder/scripts/tokenizer/tokenizer.perl -l en < corpus/FrEn/training.fr > corpus/FrEn/training.tok.fr
73  mosesdecoder/scripts/tokenizer/tokenizer.perl -l fr < corpus/FrEn/training.fr > corpus/FrEn/training.tok.fr
74  mosesdecoder/scripts/tokenizer/tokenizer.perl -l en < corpus/FrEn/test.en > corpus/FrEn/test.tok.en
75  mosesdecoder/scripts/tokenizer/tokenizer.perl -l fr < corpus/FrEn/test.fr > corpus/FrEn/test.tok.fr

77  mosesdecoder/scripts/recaser/train-truecaser.perl --model corpus/FrEn/truecase-model.en --corpus corpus/FrEn/training.tok.en
78  mosesdecoder/scripts/recaser/train-truecaser.perl --model corpus/FrEn/truecase-model.fr --corpus corpus/FrEn/training.tok.fr
79  mosesdecoder/scripts/recaser/truecase.perl --model corpus/FrEn/truecase-model.en --corpus corpus/FrEn/training.tok.en
80  mosesdecoder/scripts/recaser/truecase.perl --model corpus/FrEn/truecase-model.en < corpus/FrEn/training.tok.en > corpus/FrEn/training.tok.true.en
81  mosesdecoder/scripts/recaser/truecase.perl --model corpus/FrEn/truecase-model.fr < corpus/FrEn/training.tok.fr > corpus/FrEn/training.tok.true.fr
```

```
                              83  mosesdecoder/scripts/training/clean-corpus-n.perl
corpus/FrEn/training.tok.true fr en corpus/FrEn/training.clean 1 80

                              84  mosesdecoder/scripts/recaser/truecase.perl --model
corpus/FrEn/truecase-model.en < corpus/FrEn/test.tok.en > corpus/FrEn/test.tok.true.en
                              85  mosesdecoder/scripts/recaser/truecase.perl --model
corpus/FrEn/truecase-model.fr < corpus/FrEn/test.tok.fr > corpus/FrEn/test.tok.true.fr
                              86  mosesdecoder/scripts/training/clean-corpus-n.perl
corpus/FrEn/test.tok.true fr en corpus/FrEn/test.clean 1 80
                              ================

          === TAO LM ===
          cd lm
          88  ../mosesdecoder/bin/lmplz -o 3 < ../corpus/FrEn/training.clean.en >
training.fr-en.arpa.en
          89  echo 'is this an English sentence ?' | ../mosesdecoder/bin/query
training.fr-en.arpa.en

          91  ../mosesdecoder/bin/build_binary training.fr-en.arpa.en training.fr-en.blm.en
                    # tao binarising de truy van nhanh gon hon ?!
          92  clear
          93  echo 'is this an English sentence ?' | ../mosesdecoder/bin/query training.fr-en.blm.en
          # test truy van dich theo mo hinh ngon ngu binarised
          94  cd ../working/

          ## SRILM
          cd srilm/bin/i686-m64/
          ./ngram-count -order 4 -text ~/corpus/FrEn/training.clean.en -lm
~/lm/training.fr-en.srilm.arpa.en

          ~/mosesdecoder/bin/build_binary training.fr-en.srilm.arpa.en training.fr-en.srilm.blm.en

          **************

          ==============

          === TRAINING ===

          # training with GIZA++
          96  nohup nice ~/mosesdecoder/scripts/training/train-model.perl -root-dir train -corpus
~/corpus/FrEn/training.clean -f fr -e en -alignment grow-diag-final-and -reordering
msd-bidirectional-fe -lm 0:3:$HOME/lm/training.fr-en.blm.en :8 -external-bin-dir
~/mosesdecoder/tools >& training.out &
```

```
# training with MGIZA :: chu y ve nguon ra background :: >& training.out &
112  nohup nice ~/mosesdecoder/scripts/training/train-model.perl -root-dir train -corpus
~/corpus/FrEn/training.clean -f fr -e en -mgiza -reordering msd-bidirectional-fe -lm
0:3:$HOME/lm/training.fr-en.blm.en:8 -external-bin-dir ~/mosesdecoder/tools > training.out
```

== DICH ==

```
113  nohup nice ~/mosesdecoder/bin/moses -f ~/working/train/model/moses.ini <
~/corpus/FrEn/test.clean.fr > ~/working/test.translated.en 2> ~/working/test.out
```

=============

=== EVALUATION OF BLEU ===

```
114  ~/mosesdecoder/scripts/generic/multi-bleu.perl -lc ~/corpus/FrEn/test.clean.en <
~/working/test.translated.en > bleu.out
```

==============

Experimentations ::

FR-EN :

20k
training set = 80% = 18k
dev set = 10% = 1k
test set = 10% = 1k

cleaning : cutoff 1-80 ratio=9
training set = 17.803 sentences pairs
test set = 992 sentences pairs

Time of process : 15 minutes

```
BLEU = 19.25, 54.1/25.0/13.3/7.6 (BP=1.000, ratio=1.113, hyp_len=27863,
ref_len=25035)
```