

Moses 安装训练全过程

清华大学语音与语言技术中心 骆天一

我是以 root 权限在 Ubuntu 服务器上安装运行的 Moses，总结了网上的一些经验，再加上自己遇到一些问题然后解决问题的过程，跟大家分享一下。我安装运行成功所采用的 Ubuntu、Boost、IRSTLM 和 giza++ 的版本组合如下：Ubuntu 12.04.1 LTS + Boost_1_55_0 + irstlm-5.80.08 + giza-pp-v1.0.7。

一、Boost、GIZAA++、IRSTLM 的安装

在安装之前需要进行以下软件的安装（Moses 的官方上有）：

```
sudo apt-get install build-essential git-core pkg-config automake libtool wget zlib1g-dev python-dev libbz2-dev libsoap-lite-perl
```

1、Boost 的安装

Moses 的官方文档上说，在 Ubuntu 12.04 上 Boost 有 broken versions，所以必须自己下载安装（我将其安装在/home/lty/Moses/boost_1.55_0 目录中，Moses 是我创建的一个目录总体进行 Moses 的运行）：

```
wget http://downloads.sourceforge.net/project/boost/boost/1.55.0/boost_1_55_0.tar.gz
tar xzvf boost_1_55_0.tar.gz
cd boost_1_55_0/
./bootstrap.sh
./b2 -j4 --prefix=/home/lty/Moses/boost_1_55_0 --libdir=/home/lty/Moses/boost_1_55_0/lib64 -
-layout=system link=static install || echo FAILURE
```

2、GIZAA++ 的安装

```
wget http://giza-pp.googlecode.com/files/giza-pp-v1.0.7.tar.gz
tar xzvf giza-pp-v1.0.7.tar.gz
cd giza-pp
Make
```

在编译后会生成三个可执行文件：

```
giza-pp/GIZAA++-v2/GIZAA++
giza-pp/GIZAA++-v2/snt2cooc.out
giza-pp/mkcls-v2/mkcls
```

这些文件需要放在一个文件夹当中，我将这些文件放在了/home/lty/Moses/giza-pp 中。

但是目前为了加快速度，大家都会采用 mgiza，

git clone <https://github.com/moses-smt/mgiza><https://github.com/moses-smt/mgiza>

唯一与上面不一样的地方是要多拷贝一个文件 merge_alignment.py，我将这些文件放在了/home/lty/Moses/mgiza/mgizapp/bin 中

3、IRSTLM 的安装

```
下载地址：http://sourceforge.net/projects/irstlm/
tar xzvf irstlm-5.80.08.tgz
cd irstlm-5.80.08
./regenerate-makefiles.sh
./configure --prefix=/home/lty/Moses/irstlm-5.80.08
make install
```

此时需要记住上面三个文件夹，我本机的目录是：

```
/home/lty/Moses/irstlm-5.80.08
```

```
/home/lty/Moses/boost_1_55_0（可指定，也可不指定）
```

```
/home/lty/Moses/mgiza/mgizapp/bin
```

4、Moses 下载安装

需要先下载安装一下软件：

```
sudo apt-get install git build-essential libz-dev libbz2-dev
```

然后下载 Moses：

```
git clone https://github.com/moses-smt/mosesdecoder.git
```

```
cd mosesdecoder
```

```
./bjam -j4 --with-irstlm=/home/lty/Moses/irstlm-5.80.08 --with-
```

```
giza=/home/lty/Moses/mgiza/mgizapp/bin --with-boost=/home/lty/Moses/boost_1_55_0
```

-j4 是利用 CPU 是 4 核的进行编译

二、预料预处理

1、预料的预处理 在/home/Moses/建立一个 corpus 来存放学习集，官方网站下载学习资料

```
cd
```

```
mkdir corpus
```

```
cd corpus
```

```
wget http://www.statmt.org/wmt13/training-parallel-nc-v8.tgz
```

```
tar zxvf training-parallel-nc-v8.tgz
```

1) tokenisation:在预料的单词和单词之间或者单词和标点之间插入空白，然后进行后续操作。

```
/home/lty/Moses/mosesdecoder/scripts/tokenizer/tokenizer.perl -l en < training/news-commentary-v8.fr-en.en > news-commentary-v8.fr-en.tok.en
```

```
/home/lty/Moses/mosesdecoder/scripts/tokenizer/tokenizer.perl -l fr < training/news-commentary-v8.fr-en.fr > news-commentary-v8.fr-en.tok.fr
```

2) Truecaser: 提取一些关于文本的统计信息

```
/home/lty/Moses/mosesdecoder/scripts/recaser/train-truecaser.perl --model truecase-model.en --corpus news-commentary-v8.fr-en.tok.en
```

```
/home/lty/Moses/mosesdecoder/scripts/recaser/train-truecaser.perl --model truecase-model.fr --corpus news-commentary-v8.fr-en.tok.fr
```

3) truecasing: 将语料中每句话的字和词组都转换为没有格式的形式，减少数据稀疏性问题。

```
/home/lty/Moses/mosesdecoder/scripts/recaser/truecase.perl --model truecase-model.en < news-commentary-v8.fr-en.tok.en > news-commentary-v8.fr-en.true.en
```

```
/home/lty/Moses/mosesdecoder/scripts/recaser/truecase.perl --model truecase-model.fr <
news-commentary-v8.fr-en.tok.fr > news-commentary-v8.fr-en.true.fr
```

4)cleaning: 将长语句和空语句删除, 并且将不对齐语句进行处理。

```
/home/lty/Moses/mosesdecoder/scripts/training/clean-corpus-n.perl news-commentary-v8.fr-
en.true fr en news-commentary-v8.fr-en.clean 1 80
```

三、语言模型、机器翻译模型训练、Tuning 翻译模型和测试

1、先要进行语言模型训练

语言模型(LM)用于确保流利的输出, 在这一步使用 Iirstlm 进行处理。

```
/home/lty/Moses/mosesdecoder/scripts/generic/trainlm-irst2.perl -cores 4 -irst-dir
/home/lty/Moses/irstlm-5.80.08/bin -p 0 -order 5 -text news-commentary-v8.fr-en.true.en -lm
small-news-commentary-v8.fr-en.blm.en
```

```
echo "is this an English sentence" | /home/lty/Moses/mosesdecoder/bin/query news-
commentary-v8.fr-en.blm.en
```

2、翻译模型的训练

```
cd ..
```

```
mkdir working
```

```
cd working
```

```
/home/lty/Moses/mosesdecoder/scripts/training/train-model.perl -root-dir train -corpus
/home/lty/Moses/corpus/small-news-commentary-v8.fr-en.clean -f fr -e en -alignment grow-diag-
final-and -reordering msd-bidirectional-fe -lm 0:3:/home/lty/Moses/corpus/small-news-
commentary-v8.fr-en.blm.en:8 -external-bin-dir /home/lty/Moses/giza-pp
```

3、Tuning 翻译模型

回到 corpus, 下载开发集

```
wget http://www.statmt.org/wmt12/dev.tgz
```

```
tar zxvf dev.tgz
```

```
/home/lty/Moses/mosesdecoder/scripts/tokenizer/tokenizer.perl -l en < dev/news-test2008.en >
news-test2008.tok.en
```

```
/home/lty/Moses/mosesdecoder/scripts/tokenizer/tokenizer.perl -l fr < dev/news-test2008.fr >
news-test2008.tok.fr
```

```
/home/lty/Moses/mosesdecoder/scripts/recaser/truecase.perl --model truecase-model.en <
news-test2008.tok.en > news-test2008.true.en
```

```
/home/lty/Moses/mosesdecoder/scripts/recaser/truecase.perl --model truecase-model.fr < news-
test2008.tok.fr > news-test2008.true.fr
```

开发集在进行了和学习集相同的处理之后, 对原本的 moses.ini 进行调优

进入 working 文件夹然后运行

```
nohup /home/lty/Moses/mosesdecoder/scripts/training/mert-moses.pl
/home/lty/Moses/corpus/news-test2008.true.en /home/lty/Moses/corpus/news-test2008.true.fr
/home/lty/Moses/mosesdecoder/bin/moses train/model/moses.ini --mertdir
/home/lty/Moses/mosesdecoder/bin/ --decoder-flags="-threads 8" 2>&1 1> Result20160120 &
```

4、测试模型

如果有单独的测试集, 也要进行和开发集及训练集一样的预处理, 这里用开发集直接进行了简单测试

```
/home/lty/Moses/mosesdecoder/bin/moses -f /home/lty/Moses/working/mert-work/moses.ini  
< small-news-test2008.true.fr > ewstest2011.out
```

为了加快速度, 需要把 phrase-table 和 reordering-table 进行优化加快速度, 关于 phrase-table, 请参考 <http://www.statmt.org/moses/?n=Advanced.RuleTables#ntoc3> 中的 On-Disk Phrase table 部分。关于 reordering-table, 请参考以下命令执行:

```
cd /home/lty/Moses/corpus/mert-work  
mkdir binarised-model  
/home/lty/Moses/mosesdecoder/bin/CreateOnDiskPt      1      1      4      100     2  
/home/lty/Moses/corpus/train/model/phrase-table.gz  /home/lty/Moses/corpus/mert-work  
binarised-model/phrase-table.1.folder  
/home/lty/Moses/mosesdecoder/bin/processLexicalTable -in  
/home/lty/Moses/corpus/train/model/reordering-table.wbe-msd-bidirectional-fe.gz -out  
/home/lty/Moses/corpus/mert-work/binarised-model/reordering-table
```

然后要修改/home/lty/Moses/corpus/mert-work 中的 mose.ini 文件, 修改成:

```
PhraseDictionaryOnDisk      name=TranslationModel0      num-features=4  
path=/home/lty/Moses/corpus/mert-work/binarised-model /phrase-table.1.folder input-factor=0  
output-factor=0  
LexicalReordering name=LexicalReordering0 num-features=6 type=wbe-msd-bidirectional-fe-allff  
input-factor=0 output-factor=0 path=/home/lty/Moses/corpus/mert-work/binarised-model  
/reordering-table
```

经过这个操作之后, 生成速度加快了 45 倍, ^_^。