简介

Moses是由英国爱丁堡大学、德国亚琛工业大学等8家单位联合开发的一个基于短语的**统计机器翻译系统**。整个系统用C++语言写成，从**训练training到解码 decoder**完全开放源代码，可以运行在Linux平台和Windows平台。是目前最流行的基于短语的机器翻译系统。

语料

b.cn ：中文语料（不要太少，最少也要上千句！笔者用了3000多句做实验）

b.en ：拼音语料（与中文语料一一对应）

注：

1. 语料最好是分词之后的，因为笔者用未分词的语料进行出错，目前还没找到为什么出错！
2. 语料一定要是utf-8编码的！

前期准备

安装工具 (带*不是必须安装，他们是可视化)

格式：`sudo apt-get install [package name]`

查看是否安装某工具包：`dpkg -l | grep [package name]`

1. g++
2. git
3. subversion
4. automake
5. libtool
6. zlib1g-dev (注：g前是数字1)
7. libboost-all-dev
8. libbz2-dev
9. liblzma-dev
10. python-dev
11. make

* 12. graphviz

* 13. imagemagick

* 14. libgoogle-perftools-dev (for tcmalloc)

环境配置

1. Ubuntu
2. boost (http://downloads.sourceforge.net/project/boost/boost/1.55.0/boost*1*55_0.tar.gz)
3. irstlm5.80.08 (http://sourceforge.net/projects/irstlm/?source=navbar)
4. moses (https://github.com/moses-smt/mosesdecoder)
5. giza++ (https://github.com/jinyeqiong/BasedNLP/blob/master/GIZA%2B%2B实验.md)

6. cmph

安装Boost

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

安装GIZA++

详情可参考https://github.com/jinyeqiong/BasedNLP/blob/master/GIZA%2B%2B实验.md

```
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
```

编译后将得到的 giza-pp/GIZA++-v2/GIZA++, giza-pp/GIZA++-v2/snt2cooc.out 和 giza-pp/mkcls-v2/mkcls 三个文件放在一个文件夹中（tools），方便后期使用，笔者将该文件夹tools放在mosesdecoder（后面会讲到）下。

安装IRSTLM

下载地址：http://sourceforge.net/projects/irstlm/?source=navbar（附件即有，笔者用的是最新版本5.80.08）

详情可以看看这个网址，笔者觉得写得不错~http://blog.csdn.net/lqj1990/article/details/47105691

```
tar zxvf irstlm-5.80.08.zip
cd irstlm-5.80.08/trunk
sh regenerate-makefiles.sh
./configure --prefix=$HOME/workspace/Moses/irstlm-5.80.08 #文件所在的绝对路径
make
make install
```

安装cmph

压缩包附件即有

```
cd cmph-2.0
./configure --prefix=$HOME/workspace/Moses/cmph-2.0 #文件所在的绝对路径
make
make install
```

编译成功后，会出现一些二进制文件和脚本文件，后面要用到build_lm.sh、add-start-end.sh、compile-lm等脚本。

安装编译Moses

下载地址：https://github.com/moses-smt/mosesdecoder

```
cd mosesdecoder
./bjam --with-irstlm=/home/xuexin/workspace/Moses/itstlm-5.80.08 \
--with-mgiza=/home/xuexin/workspace/Moses/giza-pp \
--with-cmph=/home/xuexin/workspace/Moses/cmph-2.0
```

注：“\” 表示换行，没写完

--with-irstlm 用于指定irstlm的位置

--with-mgiza 用于指定mgiza++的位置

--with-cmph 用于指定cmph的位置

-j 4 用于指定核心数

等待几分钟以后，屏幕提示“success“即为编译完成 。可查看./bin目录下会有可执行文件moses。

别忘了之前将tools文件夹放在mosesdecoder文件夹下

更改环境变量

```
#在根目录下
vi .bashrc
```

进入编辑模式，输入：

```
IRSTLM="\$HOME/workspace/Moses/itstlm-5.80.08"
export IRSTLM
MOSE="\$HOME/workspace/Moses/mosesdecoder"
export MOSE
MOSE_SCRIPTS="\$MOSE/scripts"
export MOSE_SCRIPTS
PATH="\$MOSE/bin:\$MOSE_SCRIPTS/training:\$MOSE_SCRIPTS/tokenizer:\$MOSE_SCRIPTS/recaser:\$IRSTLM/bin:\$PATH"
export PATH
```

esc键退出插入状态，`:wq`退出并保存.bashrc文件。重启terminal，环境变量生效。可以利用$echo $PATH查看环境变量是否成功，在PATH内的路径下的脚本文件都可以直接使用。

至此，moses编译安装完成。

测试运行moses

```
cd ~/workspace/Moses/mosesdecoder
wget http://www.statmt.org/moses/download/sample-models.tgz
tar xzf sample-models.tgz
cd sample-models
#运行moses进行解码
../bin/moses -f phrase-model/moses.ini < phrase-model/in > out
```

如果一切正常，在out文件中有输入：this is a small house"

训练

```
cd ~/workspace/Moses
mkdir test
cd test
#将汉语(b.cn)和拼音(b.en)的语料放入test中
```

构建语言模型

语料预处理：

```
#在test文件夹下
#tokenizer：在语料的单词和单词之间或者单词和标点之间插入空白
tokenizer.perl -l en < b.en > b.tok.en -threads 8
tokenizer.perl -l en < b.cn > b.tok.cn -threads 8

#truecaser：提取一些关于文本的统计信息
train-truecaser.perl --corpus b.tok.en --model b.model.en
train-truecaser.perl --corpus b.tok.cn --model b.model.cn

#truecasing：将语料中每句话的字和词组都转换为**没有格式**的形式，减少数据的稀疏性问题
truecase.perl --model b.model.en < b.tok.en > b.true.en
truecase.perl --model b.model.cn < b.tok.cn > b.true.cn
```

```
#将长语句和空语句删除，并且将不对齐语句进行处理
clean-corpus-n.perl b.true cn en b.clean 1 80
```

语言模型训练：

```
#对语料加首尾标识符
add-start-end.sh < b.clean.cn > b.sb.cn
add-start-end.sh < b.clean.en > b.sb.en

#生成语言模型文件
build-lm.sh -i b.sb.cn -t ./tmp -p -s improved-kneser-ney -o b.lm.cn
build-lm.sh -i b.sb.en -t ./tmp -p -s improved-kneser-ney -o b.lm.en

#将文件转换成标准的ARPA格式
compile-lm --text b.lm.cn.gz b.arpa.cn
compile-lm --text b.lm.en.gz b.arpa.en

#使用KenLM对其进行二值化，为了让程序更快载入
build_binary b.arpa.cn b.blm.cn
build_binary b.arpa.en b.blm.en
```

但是很不幸，我运行很多遍，都不能运行统，生成的ARPA格式一直都是slatin1格式。后来请教高人，决定换一种方法，采用KenLM方法得到ARPA格式文件，并进行二值化，下面是具体过程（代替以上代码后三步！！！）：

详情参考：http://kheafield.com/code/kenlm/

```
wget -O - http://kheafield.com/code/kenlm.tar.gz | tar xz
cd kenlm
./bjam -j4

#生成ARPA格式文件
bin/lmplz -o 5 < b.sb.cn > b.arpa.cn
bin/lmplz -o 5 < b.sb.en > b.arpa.en

#进行二值化
bin/build_binary b.arpa.cn b.lm.cn
bin/build_binary b.arpa.en b.lm.en
```

构建翻译模型

```
#在Moses文件夹下
mkdir working
cd working
nohup nice train-model.perl -root-dir train -corpus ../test/b.clean -f cn -e en -alignment grow-diag-final-and -reordering msd-bid
```

注意：-lm 参数后面的要用绝对地址！！！

结束后，working文件夹下会出现train目录，在train/model/能找到配置文件moses.ini。

测试训练模型是否正确

```
#在working文件夹下
echo "我是小明" | ~/workspace/Moses/mosesdecoder/bin/moses -f train/model/moses.ini > out
cat out
"I'm xiaoming"
```

调优

相信大家在实验环节就能注意到，只要输入了语料以外的语法，翻译就不准确了，因此我们需要使用有限的语料库，实现高质量的翻译。

train-model.perl默认是Giza++完成词对齐处理（占整体训练时间的70%），但是它只支持单线程处理；因此需要MGiza工具，实现多线程处理。

```
#优化脚本("\"代表一行没写下，换行写，但是输入时是在一行！)
nohup nice train-model.perl -mgiza -mgiza-cpus 8 -cores 8 -parallel -sort-buffer-size 6G \
-sort-batch-size 400 -sort-compress gzip -sort-parallel 10 -root-dir train -corpus ../b.clean -f cn -e en \
-alignment grow-diag-final-and -reordering msd-bidirectional-fe \
-lm 0:3:/home/xuexin/workspace/Moses/test/b.lm.en:8 \
-external-bin-dir /mgiza ~/workspace/Moses/mosesdecoder/tools >& training.out &
```

【注：MGiza没有安装上，还有待研究！！！】

使用优化好的脚本，理论上性能可以提升80%，但还得看实际硬件，CPU是i5以下，将所有并发数降低到4。

-mgiza-cpus 8 -cores 8，一个指的是对齐统计时的并发数，一个指的是文件输出时的并发数，它俩在执行时有先后顺序，并不冲突。

注：

1. 该指标在虚拟机上不准确，虚拟机实际性能要看其实体机的共享资源而定。
2. 训练过程会将磁盘文件按堆拷贝到内存中，反复多次迭代，所以磁盘的性能也会产生很大影响，如果有条件，fusionIo或ssd将会带来很好的性能提升。

参考文章：

1. http://wenku.baidu.com/link?url=ds_CZNtj-QhTZNxfvlUpRGhtHuviBTL4mAGCLO6lULMLZ1Pas5rf99bcEX7-KWz4mjMnjo9fCgZijFtkNJPf0cB4-bdoHXvCWvNDDFaGx6i
2. http://wenku.baidu.com/link?url=mw3GKklFd4TCZt_O2e3vfvrUuGxhmu7SoBqt04nD6BwLgWRlVsJ5lS1hammjmjri3uSeN8c52eKB89tXJ1legiyZCybY51S3OfumCtMRu8q 本人觉得不错~
3. http://www.cnblogs.com/panweishadow/p/4771050.html
4. http://blog.csdn.net/lqj1990/article/details/47105691
5. http://blog.csdn.net/lqj1990/article/details/47083067
6. http://kheafield.com/code/kenlm/