Moses 操作手册

冯洋

2016-12-15

1. 编译

将 Moses 文件夹存放在某个目录下，进入 Moses 目录，在命令行输入以下命令：

```
export MOSES=$PWD
```

1.1 依赖

1.1.1 gcc

请保证 gcc 的版本不要太低，最好是 gcc4.8

1.1.2 boost

boost 版本需要至少是 1.48，我用的是 1.62

export $BOOST=$MOSES/usr/boost

```
./bootstrap.sh
./b2 --prefix=$BOOST --libdir=$BOOST/lib64 --layout=tagged
link=static,shared threading=multi install
```

1.1.3 bzip2

需安装 bzip2，请参考 http://www.bzip.org/

修改~/.bashrc 的 PATH, LD_LIBRARY_PATH，以包括以上工具的 bin 和对应的 lib 或者 lib64 目录。

1.2 SRILM

进入$MOSES/srilm 目录。

首先修改 Makefile 文件的 SRILM 的路径，改成为现在 srilm 的根目录，为

```
SRILM = $MOSES/srilm
```

最后运行命令

```
make World
```

至此，srilm 编译完成，检查$MOSES/srilm/bin/i686-m64/ ngram-count 是否可以运行，如果可以运行，证明编译成功。

1.3 GIZA++

进入$MOSES/giza-pp 目录，运行 make 直接进行编译，检查 GIZA++-v2 目录下是否有以下可执行文件：

```
GIZA++, plain2snt.out, snt2cooc.out, snt2plain.out
```

将这些可执行文件拷贝到 giza-pp 目录下。

1.4 Moses

进入$MOSES/mosesdecoder 目录，运行以下命令进行编译：

```
./bjam -aq -j 4 --with-boost=$BOOST--with-srilm-=$MOSES/srilm
```

如果编译成功，在$MOSES/mosesdecoder/bin 目录下有 moses 和 moses_chart 可执行程序，如果运行会输出相应的命令选项。

2. 运行

2.1 数据准备

创建以下目录

```
$MOSES/workplace,
$MOSES/workplace/data,
$MOSES/workplace/dev_test
```

在该总目录下新建一个目录 data 用来存放开发集、测试集以及原始的语料。

开发集和测试集都是一个句子一行，不加任何标记，对于参考译文，一个参考译文一个文件，后缀从 0 开始算起。这里开发集选定 nist02，测试集选定 nist05，所以用到的文件为 nist02_src，nist05_src，nist02_ref0，…，nist02_ref3，nist05_ref0， …，nist05_ref3。

语言模型假设已经训练好了，为

```
$MOSES/lm/target.o5.lm.gz
```

2.1 抽取短语

2.1.1 从头开始执行

语料准备

训练语料，源语言和目标语言分别放在一个文件里，一行一个句子，

可以加 <s> 和 </s> 。这里用到的是 NIST09_FBIS.cn 和 NIST09_FBIS.en，由于 GIZA++要求句子长度不超过 100，所以首先过滤掉句子长度大于 100 的句子

```
$HOME/moses/scripts/target/scripts-20091207-1036/training/clean-corpus-n.perl NIST09_FBIS cn en FBIS_clean 1 100
```

过滤后得到的文件为 FBIS_clean.cn FBIS_clean.en，句子的长度都小于 100.

同时需要过滤到预料中的空行以及”|”，否则 GIZA++会报错。

注意：需要查看 clean-corpus-n.perl 文件的第 8 行所设定的编码格式，默认的是 utf-8，而我们一般用的是 GBK，所以需要将第 8 行修改为

```
my $enc = "gbk"; # encoding of the input and output files
```

抽取短语

```
$HOME/Moses/mosesdecoder/scripts/training/train-model.perl
 --external-bin-dir $HOME/Moses/giza-pp
 --root-dir $HOME/Moses/workplace_stem/train
 --corpus $HOME/Moses/workplace_stem/mydata/training_stem
 --f source
 --e target
 --first-step 1
 --last-step 9
 --alignment grow-diag-final-and
 --reordering msd-backward-fe
 --lm 0:5:$HOME/Moses/workplace_stem/lm/target.o5.lm.gz
```

得到的结果存放在 model 目录下：phrase-table.0-0.gz，reordering-table.0-0.gz，generation.0-0.gz。

抽取短语完成之后，会在 model 目录下生成一个 moses.ini 文件，用来进行 mert 训练时的解码，这个文件的第 18-19 行设计到 generation 文件

```
[generation-file]
0 0 2 /home3/ly/fy/moses_data6/model/generation.0-0
```

由于我们上面 generation 的要素设为 0-0，所以不需要这个配置，必须注释掉，否则会出错。

2.1.2 已有 GIZA++对齐结果

语料准备

抽取出 GIZA++的结果，源语言文件，目标语言文件，对齐文件，都是每行一句，没有标签。

注意：如果 GIZA++加了<s>和</s>，抽取的时候不能去掉，否则对齐就会错位。

将这三个文件复制到 model 目录下，分别命名 aligned.0.cn，aligned.0.en，aligned.grow-diag-final，同时需要将 aligned.0.cn 和 aligned.0.en 复制到 model 目录下，改名为 FBIS_clean.cn，FBIS_clean.en。

抽取短语

这里需要加上--first-step 4.

```
/home3/ly/fy/moses/scripts/target/scripts-20091207-1036/training/train-factored-phrase-model.perl –first-step 4 --scripts-root-dir /home3/ly/fy/moses/ scripts/target/scripts-20091207-1036 --corpus ./data/FBIS_clean --f cn --e en --max-phrase-length 7 --alignment-factors 0-0 --translation-factors 0-0 --reordering msd-fe --reordering-factors 0-0 --generation-factors 0-0 --lm 0:5:/home3/ly/fy/data /afp.xh.v2.sri.kn.o5.lm.gz
```

得到的结果存放在 model 目录下：phrase-table.0-0.gz，reordering-table.0-0.gz，generation.0-0.gz。

抽取短语完成之后，会在 model 目录下生成一个 moses.ini 文件，用来进行 mert 训练时的解码，这个文件的第 18-19 行设计到 generation

文件

2.2 mert 训练

不管是否用已有 GIZA++结果，该步骤都相同。

```
/home3/ly/fy/moses/scripts/target/scripts-20091207-1036/training/mert-moses.pl --input ./data/nist02_src –refs ./data/nist02_ref --decoder /home3/ly/fy/moses/moses-cmd/src/moses --config ./model/moses.ini --rootdir /home3/ly /fy/moses/scripts/target/scripts-20091207-1036 --working-dir . --nbest 100
```

如果中断了，要继续进行 mert 训练，只需要添加选项—continue.

训练的时候会默认的对短语表和重排序表进行过滤，实际用到的是 filetered 目录下的 moses.ini 和 phrase-table.0-0.1，reordering-table.0-0。

2.4 测试

对于大语料，必须对测试集进行短语表的过滤，小语料也可以不进行过滤。

```
/home3/ly/fy/moses/scripts/target/scripts-20091207-1036/training/filter-model-given-input.pl ./filtered_test ./moses.ini ./data/nist05_src
```

过滤后的配置文件 moses.ini，短语表以及重排序表存放在目录 filtered_test 下。

测试运行以下命令

```
/home3/ly/fy/moses/moses-cmd/src/moses -f ./filtered_test/moses.ini
< ./data/nist05_src > result.txt
```

生成的结果文件 result.txt 是一个 plain 文件，一行一个结果，没有进行后处理，如果需要测试最终结果的 BLEU 值，还需要运行

```
./bin/plain2sgm_c2e result.txt ./data/nist05_src.sgm result.sgm
./bin/bproc_c2e -l -o result.sgm -n result.bproc
./bin/mteval-v11b.pl -c -r ./data/nist05_ref.sgm
-s ./data/nist05_src.sgm -t result.bproc > result.bproc.eval
```

3. 添加命名实体识别