MOSES安装记录

环境: Ubuntu16.04

需要安装的工具: giza++,IRSTLM5.80.08,cmph2.0,xmlrpc1.33.14,boost1.64

o安装依赖的包

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

1 从GitHub上clone moses

- 1 cd ~
- 2 mkdir smt
- 3 cd smt
- 4 git clone https://github.com/moses-smt/mosesdecoder.git

2 安装giza++

- 1 git clone https://github.com/moses-smt/giza-pp.git
- 2 cd giza-pp
- 3 make

把之后需要用到的giza++文件复制到mosedecoder文件夹中创建的tools文件夹下

- 1 cd ../mosesdecoer
- 2 mkdir tools
- 3 cp ../giza-pp/GIZA++-v2/GIZA++ ../giza-pp/GIZA++-v2/snt2cooc.out ../giza-pp/mkcls-v2/mkcls tools

3 安装IRSTLM5.80.08

- 1 cd ..
- 2 mkdir irstlm
- 3 wget https://jaist.dl.sourceforge.net/project/irstlm/irstlm/irstlm-5.80/irstlm-5.80.08.tgz
- 4 tar zxvf irstlm-5.80.08.tgz
- 5 cd irstlm-5.80.08
- 7 ./regenerate-makefiles.sh
- 8 ./configure --prefix=~/smt/irstlm #设置irstlm的安装路径

9 make install

4 安装cmph2.0

- 1 cd ~/smt
- 2 wget http://www.achrafothman.net/aslsmt/tools/cmph_2.0.orig.tar.gz
- 3 tar zxvf cmph_2.0.orig.tar.gz
- 4 cd cmph-2.0/
- 5 ./configure
- 6 make
- 7 make install

5 安装xmlrpc1.33.14

- 1 cd ~/smt
- 2 wget http://www.achrafothman.net/aslsmt/tools/xmlrpc-c_1.33.14.orig.tar.gz
- 3 tar zxvf xmlrpc-c_1.33.14.orig.tar.gz
- 4 cd xmlrpc-c-1.33.14/
- 5 ./configure
- 6 make
- 7 make install

6 安装boost1.64

```
1 cd ~/smt
2 wget https://dl.bintray.com/boostorg/release/1.64.0/source/boost_1_64_0.tar.gz
3 tar zxvf boost_1_64_0.tar.gz
4 cd boost_1_64_0/
5 ./bootstrap.sh
6 ./b2 --layout=system link=static install || echo FAILURE
```

7.1 编译moses

```
1 cd ~/smt/mosesdecoder
2 make -f contrib/Makefiles/install-dependencies.gmake
```

在这个过程出现了ERROR

```
1 sudo vim install-dependencies.gmake
```

通过查看这个gmake文件,找到问题的位置。问题应该是出在在这个文件编译过程中xmlrpc这个包下载不下来,看了一些博客和教程他们的方法都不管用,我自己认为xmlrpc之前已经安装过了,所以我注释了这一段。

```
Lirgithub = https://github.com/
Lirsourceforge = http://downloads.sourceforge.net/project
Lirgithub
BES
.00# functions for building software from sourceforge
Lirnproc := $(shell getconf _NPROCESSORS_ONLN)

Lirsfget = mkdir -p '${TMP}' && cd '${TMP}' && wget -q0- ${URL} | tar zxv

Lirconfigure-make-install = cd '$1' && ./configure --prefix='${PREFIX}'

Narconfigure-make-install += && make -j${nproc} && make install
16
ub # XMLRPC-C for moses server
h:xmlrpc: URL=$(sourceforge)/xmlrpc-c/Xmlrpc-c%20Super%20Stable/1.33.17/xmlrpc-c-1
ubuxmlrpc: TMP=$(CWD)/build/xmlrpc
ublxmlrpc: override PREFIX=${XMLRPC_PREFIX}
bocxmlrpc: | $(call safepath,${XMLRPC_PREFIX}/bin/xmlrpc-c-config)
boc$(call safepath,${XMLRPC_PREFIX}/bin/xmlrpc-c-config):
              strept.n.

$(sfget)
$(sall configure-make-install,${TMP}/xmlrpc-c-1.33.17)
rm -rf ${TMP}
CMP
ubu
ubu
/hc# CMPH for CompactPT
ubucmph: URL=$(sourceforge)/cmph/cmph/cmph-2.0.tar.gz
    cmph: TMP=$(CWD)/build/cmph
                                                                                                             62%
                                                                                         73,1
```

而且在图中可以看到xmlrpc这个包的URL地址,我直接去这个地址上也down不下来,它的版本跟我在前几步中下载安装的版本也不对。

7.2 安装moses

```
1 cd ~/smt/mosesdecoder
2 sudo ./bjam --with-boost=../boost_1_64_0 --with-cmph=../cmph-2.0 --with-irstlm=../irstlm --with-giza=../giza-pp
```

这个过程需要一点时间,我在这一步失败了,提示我built failed

```
🗎 🔳 ubuntu@ubuntu: ~/smt/mosesdecoder
ary_test.passed
Running 2 test cases...
 *** No errors detected
gcc.link meri/meri
gcc.link mert/extractor
gcc.link mert/evaluator
gcc.link mert/pro
gcc.link mert/kbmira
gcc.link mert/sentence-bleu
gcc.link mert/sentence-bleu-nbest
gcc.link mert/hgdecode
...failed updating 1 target...
...skipped 83 targets...
...updated 1160 targets...
The build failed. If you need support, run:
./jam-files/bjam --with-boost=../boost_1_64_0 --with-cmph=../cmph-2.0 --with-i
rstlm=../irstlm --with-giza=../giza-pp --debug-configuration -d2 |gzip >build.lo
g.gz
then attach build.log.gz to your e-mail.
You MUST do 3 things before sending to the mailing list:
    1. Subscribe to the mailing list at http://mailman.mit.edu/mailman/listinfo/m
oses-support
    2. Attach build.log.gz to your e-mail
```

运行如下的命令

```
sudo ./jam-files/bjam --with-boost=../boost_1_64_0 --with-cmph=../cmph-2.0 --with-irstlm=../irstlm --with-giza=../giza-pp --debug-configuration -d1 |gzip >build.log.gz
```

在mosesdecoder文件中找到build.log.gz这个压缩包

```
1 #解压缩
2 gzip -d build.log.gz
```

```
🔊 🖨 📵 ubuntu@ubuntu: ~/smt/mosesdecoder
  ubuntu@ubuntu:~/smt/mosesdecoder$ ls
azure-pipelines.yml
                                           lib
                                                       opt
                         COPYING
□bin
                                                       previous.sh
 bjam
build
                                                       README
                         doxygen.conf
 cgmanifest.json
                         env-check.yml
                                                       run-regtests.sh
 chk.tmp
                                                      sample-models
 compile.sh
                         Jamroot
Oubuntu@ubuntu:~/smt/mosesdecoder$ gzip -d build.log.gz
ubuntu@ubuntu:~/smt/mosesdecoder$ ls
 azure-pipelines.yml
                                          lib
                                                      opt
                         COPYING
 bin
                                                       previous.sh
 bjam
 build
                                                       README
 build.log
                         doxygen.conf
  cgmanifest.json
                         env-check.yml
                                                       run-regtests.sh
 chk.tmp
                                                       sample-models
  compile.sh
                         Jamroot
 ubuntu@ubuntu:~/smt/mosesdecoder$
```

查看build.log

1 vim build.log

```
Unable to load Boost.Build: could not find "boost-build.jam"

Attempted search from /home/ubuntu/smt/mosesdecoder up to the root at /home/ubuntu/smt/mosesdecoder/share/boost-build and in these directories from BOOST_BUILD_PATH and BOOST_ROOT: /usr/share/boost-build.
Please consult the documentation at 'http://www.boost.org'.
```

提示找不到boost-build.jam文件,这个问题解释起来比较麻烦,我通过在.bashrc文件中加入路径解决了这个问题。

```
1 # .bashrc文件在用户目录下
2 cd ~
3 vim .bashrc
```

在文件的最后添加上以下几句

```
1 IRSTLM = "$HOME/smt/irstlm"
2 export IRSRLM
3 MOSE = "$HOME/smt/mosesdecoder"
4 export MOSE
5 MOSE_SCRIPTS = "$MOSE/scripts"
6 export MOSE_SCRIPTS
7 PATH="$MOSE/bin:$MOSE_SCRIPTS/training:$MOSE_SCRIPTS/tokenizer:$MOSE_SCRIPTS/recaser:$IRSTLM/bin:$PATH"
8 export PATH
```

```
😑 💷 ubuntu@ubuntu: ~
     . ~/.bash_aliases
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
   if [ -f /usr/share/bash-completion/bash_completion ]; then
  . /usr/share/bash-completion/bash_completion elif [ -f /etc/bash_completion ]; then
     . /etc/bash_completion
IRSTLM = "$HOME/smt/irstlm"
export IRSRLM
MOSE = "$HOME/smt/mosesdecoder"
export MOSE
MOSE_SCRIPTS = "$MOSE/scripts"
export MOSE_SCRIPTS
PATH="$MOSE/bin:$MOSE_SCRIPTS/training:$MOSE_SCRIPTS/tokenizer:$MOSE_SCRIPTS/rec
aser:$IRSTLM/bin:$PATH"
export PATH
                                                                             118,0-1
                                                                                               But
```

保存退出后再次进行安装moses

```
1 cd ~/smt/mosesdecoder
2 sudo ./bjam --with-boost=../boost_1_64_0 --with-cmph=../cmph-2.0 --with-irstlm=../irstlm --with-giza=../giza-pp
```

```
🙆 🖨 💷 ubuntu@ubuntu: ~/smt/mosesdecoder
ntenceOnDemandLM.o): In function `Moses::InMemoryPerSentenceOnDemandLM::Initiali
zeForInput(boost::shared_ptr<Moses::TranslationTask> const&)':
InMemoryPerSentenceOnDemandLM.cpp:(.text+0x9a3): warning: the use of `tmpnam' is
 dangerous, better use `mkstemp'
testing.unit-test moses/bin/gcc-5.4.0/release/link-static/threading-multi/moses_
test.passed
Running 21 test cases..
Constructing a Sparse Reordering feature
*** No errors detected
gcc.link moses/LM/bin/BackwardTest.test/gcc-5.4.0/release/link-static/threading-
multi/BackwardTest
moses/LM/bin/BackwardTest.test/gcc-5.4.0/release/link-static/threading-multi/InMemoryPerSentenceOnDemandLM.o: In function `Moses::InMemoryPerSentenceOnDemandLM:
:InitializeForInput(boost::shared_ptr<Moses::TranslationTask> const&)'
InMemoryPerSentenceOnDemandLM.cpp:(.text+0x9a3): warning: the use of `tmpnam' is
dangerous, better use `mkstemp' `
testing.capture-output moses/LM/bin/BackwardTest.test/gcc-5.4.0/release/link-sta
tic/threading-multi/BackwardTest.run
**passed** moses/LM/bin/BackwardTest.test/gcc-5.4.0/release/link-static/threadin
g-multi/BackwardTest.test
  .undated 85 targets...
SUCCESS
ubuntuqubuntu:~/smt/mosesdecoder$
           ☐ Connect to Serve
```

7.3 测试moses

1 vim out

```
1 cd ~/mosesdecoder
2 wget http://www.statmt.org/moses/download/sample-models.tgz
3 tar xzf sample-models.tgz
4 cd sample-models
5
6 # run the decoder
7 cd ~/mosesdecoder/sample-models
8 ~/mosesdecoder/bin/moses -f phrase-model/moses.ini < phrase-model/in > out
```

在sample-models下面回生成两个文件: nbest.txt和out

可以打开看一下out文件,会显示两行this is a small house,说明安装是成功的

