Moses on Windows 7

Amittai Axelrod

v1.0, 2011.07.15

This is a guide to installing and compiling the Moses machine translation framework (stable release dated 2010-08-13) on a Windows 7 machine running Cygwin (latest release; version 1.7.9). Except when noted, each quoted line (written in `typewriter font`) goes on a separate line.

1 Install Cygwin

Download the Cygwin installer from `www.cygwin.com/setup.exe`, then double-click to install Cygwin on your computer. Install the binaries (*.bin*) of the following additional packages:

- `make`
- `g++` (currently 4.3.4)
- `autoconf`
- `automake`
- `libtool`
- `boost`
- `libboost` (newer than 1.31.0)
- `flip` (optional, but useful)

Some of these packages will also prompt you to install related or dependent packages. These additional packages are generally necessary, so just accept the defaults. Run Cygwin now for the first time, so that your home directory (`/home/`*your_username*`/`) is created.

2 Install 7zip (Optional)

Many files are in *.tar*, *.gz*, or *.tgz* format, and it may be useful to have a Windows program to handle these. It is optional, but the `7-zip` utility works well for this. Download from `www.7-zip.org/download.html` and install it.

3 Install GIZA^{++}

Download version 1.0.5 (the latest) of GIZA^{++} from http://code.google.com/p/giza-pp/.
For convenience, unpack it into a top-level directory such as ~/giza-pp/.
GIZA^{++} won't compile directly, as the __GNUC__ environment variable isn't used(?) in
Cygwin. Therefore, comment out most of lines 22-31 of ~/giza-pp/GIZA++-v2/mystl.h
to look like so:

```
//#if __GNUC__==2
//#include <hash_map>
//#elsif __GNUC__==3
#include <ext/hash_map>
using __gnu_cxx::hash_map;
//#else
//#include <tr1/unordered_map>
//#define hash_map unordered_map
//using namespace std::tr1;
//#endif
```

Version 1.0.5 of GIZA^{++} does not support coocurrence files. To fix this, change line 8 of
~/giza-pp/GIZA++-v2/Makefile to the following (all on one line):

```
CFLAGS_OPT = $(CFLAGS) -O3 -DNDEBUG -DWORDINDEX_WITH_4_BYTE
           -DBINARY_SEARCH_FOR_TTABLE -O3 -DNDEBUG -DWORDINDEX_WITH_4_BYTE
```

Furthermore, edit lines 16-20 of the same Makefile to look like so:

```
#ifeq ($(shell uname),Darwin)
    LDFLAGS =
#else
#   LDFLAGS = -static
#endif
```

Now you can compile:

```
cd ~/giza-pp/
make all
```

Copy the binaries into a top-level ~/bin/ directory for easy access:

```
mkdir -p ~/bin/
cp ~/giza-pp/GIZA++-v2/GIZA++.exe  ~/bin/
cp ~/giza-pp/GIZA++-v2/snt2cooc.out  ~/bin/
cp ~/giza-pp/mkcls-v2/mkcls.exe  ~/bin/
```

4 Install Moses

4.1 Compile Moses

The latest Moses release is dated 13th August 2010. Download the Moses Machine Trans-
lation framework from
http://sourceforge.net/projects/mosesdecoder/files/
The download contains two folders: moses/ and _MACOSX/. Delete the _MACOSX/ folder.
For convenience, unpack the files in the moses/ folder into ~/moses/. The following will
make a directory ~/bin/moses-scripts/ for the helper scripts, and compile the top-level
moses executables:

```
mkdir -p ~/bin/moses-scripts/
cd ~/moses/
./regenerate-makefiles.sh
./configure
make
```

Note that unlike installing on Linux, you don't use the `--with-srilm` (nor any other language model) flag when running `configure`.

4.2 Compile Moses helper scripts

Shuffle files around so that `make` looks in the right place:

```
cd ~/moses/scripts/
mv Makefile Makefile-Unix
ln -s MakefileWIN32 Makefile
```

4.2.1 MakefileWIN32

Edit `~/moses/scripts/MakefileWIN32` as follows:

- Set the `$TARGETDIR` and `$BINDIR` variables to where you want the binaries installed.

  ```
  TARGETDIR=/home/your_username/bin/moses-scripts
  BINDIR=/home/your_username/bin
  ```

- Edit line 66-68 of `~/moses/scripts/MakefileWIN32` to say [1]:

  ```
  @./check-dependenciesWIN32.pl "$(HOME)" "$(TARGETDIR)" "$(RELEASEDIR)" "$(BINDIR)"
  mkdir -p $(RELEASEDIR)
  xargs cp --parents -R -t $(RELEASEDIR) < ./released-filesWIN32
  ```

  instead of:

  ```
  @./check-dependencies.pl "$(HOME)" "$(TARGETDIR)" "$(RELEASEDIR)" "$(BINDIR)"
  mkdir -p $(RELEASEDIR)
  rsync -r --files-from ./released-files . $(RELEASEDIR)/
  ```

- Change *released-files* to *released-filesWIN32* in line 114:

  ```
  | grep -F -x -v -f released-filesWIN32
  ```

- Change references of *train-factored-phrase-model.perl* to *train-model.perl* as shown in line 51:

  ```
  MAIN_TRAINING_SCRIPTS_NAMES=filter-model-given-input.pl mert-moses.pl \
      train-model.perl clean-corpus-n.perl
  ```

  As well as line 69 (all on one line):

  ```
  sed 's#^my \$$BINDIR\s*=.*#my \$$BINDIR="$(BINDIR)";#' training/train-model.perl
      > $(RELEASEDIR)/training/train-model.perl
  ```

---

[1]Credit for the rsync-to-xargs switch goes to Tiago Tresoldi:
http://article.gmane.org/gmane.comp.nlp.moses.user/1631

4.2.2 check-dependenciesWIN32.pl

Change line 9 of `~/moses/scripts/check-dependenciesWIN32.pl` to:

```
if ($target_dir eq '' || !(-d $target_dir)) {
```

Instead of:

```
if ($target_dir eq '' || -z $target_dir) {
```

The reason is that the `-z` test is failing on a directory, which shouldn't happen. Better to just explicitly test for the directory's existence.

4.2.3 released-filesWIN32

Add this line inside `~/moses/scripts/released-filesWIN32`:

```
training/phrase-extract/consolidate.exe
```

Now convert the file's line endings to Unix format:

```
flip -u released-filesWIN32
```

4.2.4 lexical-reordering/

Change `~/moses/scripts/training/lexical-reordering/Makefile` to read:

```
$(CXX) score.cpp reordering_classes.o -lz -o score
```

Instead of the (nearly-identical!):

```
$(CXX) -lz score.cpp reordering_classes.o -o score
```

Then, compile.

```
cd ~/moses/scripts/training/lexical-reordering/
make all
```

4.2.5 mbr/

Add the following line to the start of `~/moses/scripts/training/mbr/mbr.cpp`:

```
#include <time.h>
```

Then compile:

```
cd ~/moses/scripts/training/mbr/
make all
```

4.2.6 memscore/

Create the makefile and then compile:

```
cd ~/moses/scripts/training/memscore/
./configure
make
```

4.3 Compile

You are now free to compile the Moses helper scripts:

```
cd ~/moses/scripts/
make release
```

Take note of the directory that's created (`/bin/moses-scripts/scripts-YYYYMMDD-HHMM`),
as you will need to set the following environment variable before each time you use the
Moses framework to train a model, as follows:

```
export SCRIPTS_ROOTDIR=/relative-path-to-homedir/bin/moses-scripts-scripts-YYYYMMDD-HHMM
```

If you're only using the stable Moses releases and thus not recompiling very often, consider
doing the following:

```
cd ~/bin/
ln -s moses-scripts/scripts-YYYYMMDD-HHMM moses-scripts-link
```

Now you can have a shorter path to pass around: `~/bin/moses-scripts-link/`

5 Language Model

Training a translation model is all well and good, but you will need a language model in
order to actually use it. As such, you will need to install a language modeling tool. The
SRILM toolkit, among others, can be used to build a language model. SRILM can be
downloaded here:
`http://www-speech.sri.com/projects/srilm/download.html`
Further instructions for running SRILM (or any other language modeling toolkit) is
beyond the current scope of this document. You must build a language model for the
language you wish to translate *into* before proceeding to the next section.

6 Using Moses

The use of Moses (and associated scripts) to train, tune, and evaluate a statistical machine
translation system is already well-documented by the fine folks who run the Workshop
on Machine Translation:
`http://statmt.org/wmt11/`
The procedure for running Moses on Win7 + Cygwin is almost exactly the same as it is in
Unix, because nearly all of the steps are Perl wrapper scripts. Refer to WMT baseline sys-
tem `http://statmt.org/wmt11/baseline.html` for a step-by-step guide with examples
for preparing data, building a language model, training a model, tuning it, and decoding.
The difference is that the decoder's executable file is `~/moses/moses-cmd/src/moses.exe`
instead of `moses/moses-cmd/src/moses`.

7 Citations

Open-source machine translation software is free to use for research purposes. Kindly
cite the authors of these tools in your bibliography, using the following publications:

- GIZA^{++}:

  ```
  Franz Josef Och, Hermann Ney.
  "A Systematic Comparison of Various Statistical Alignment Models"
  Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003.
  ```

- Moses

  ```
  Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico,
    Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer,
    Ondrej Bojar, Alexandra Constantin, Evan Herbst.
  "Moses: Open Source Toolkit for Statistical Machine Translation"
  Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session
  Prague, Czech Republic, June 2007.
  ```

- BLEU

  ```
  Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu.
  "BLEU: a method for automatic evaluation of machine translation"
  40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311318
  Philadelphia, July 2002.
  ```

- MERT

  ```
  Franz-Joseph Och.
  "Minimum Error Rate Training in Statistical Machine Translation"
  41st Annual Meeting of the Association for Computational Linguistics (ACL), pp160167,
  Sapporo, Japan, July 2003.
  ```

- SRILM

  ```
  Andreas Stolcke.
  SRILM -- An Extensible Language Modeling Toolkit.
  Proc. Intl. Conf. on Spoken Language Processing, vol. 2, pp. 901-904
  Denver, 2002.
  ```

8 Credits

This document was written and updated by Amittai Axelrod. If this is useful to you, then please drop me a note, buy me a beer, or both. Many thanks to Alisa Nguyen for testing, and comments from Hassan Sajjad.

9 Release Notes

1.00 – First release. Added bibliography.
0.84 – Clarify package dependencies for Cygwin
0.83 – MakefileWIN32 fix, run Cygwin note
0.82 – commented out "Additional Scripts" section, rewrote Moses usage
0.81 – added LM section, skeleton of citation/bibliography, Moses usage
0.80 – added many fixes from Alisa; fixed typos.