

11-731: Machine Translation

Homework Assignment #7:

Out: Wednesday, March 25, 2009

Due: 2:00 pm Wednesday, April 01, 2009

In this homework you will use MOSES¹ decoder to generate translations. You will also run minimum error-rate training (MERT) to optimize the decoder parameters. Use the setup provided at [/afs/cs.cmu.edu/project/cmt-55/lti/Courses/731/homework/HW7](http://afs/cs.cmu.edu/project/cmt-55/lti/Courses/731/homework/HW7).

Note: This homework will require more CPU time (and your time spread over a longer period), so start early!

Your tasks are:

1) Generate phrase table

This step is similar to what you did in homework 4. But this time you will use a modified version of the training script: `train-factored-phrase-model.perl`. You will run it up to step 9 until it generates a configuration file for MOSES that is used in later tasks. A 3-gram SRI language model is provided in `./lm`.

```
train-factored-phrase-model.perl --root-dir . --bin-dir ./bin --
scripts-root-dir ./moses/scripts --f es --e en --corpus corpus/training
--giza-f2e giza.es-en --giza-e2f giza.en-es --max-phrase-length 7 --lm
0:3:./lm/training.en.srilm
```

Note: If relative paths did not work use absolute paths.

2) Run MERT on a development set using default parameters

Use the configurations file `./model/moses.ini` generated in Task 1. A development set is provided in `./dev/dev.es` along with a reference translation `./dev/dev.en`.

Use the script `mert-moses-new.pl` in MOSES package to perform MERT. The default setting optimizes on BLEU metric.

```
mert-moses-new.pl --rootdir ./moses/scripts --mertdir ./moses/mert/
--input ./dev/dev.es --refs ./dev/dev.en --decoder ./moses/moses-
cmd/src/moses --config ./model/moses.ini >& mert.log
```

¹ <http://www.statmt.org/moses/>

MERT will run for several iterations. The intermediate files will be in `./mert-work`. At the end of the run, it will produce optimized weights for model parameters (`weights.txt`) and a new configuration file with optimized weights (`moses.ini`).

Observe how BLEU score increases over the iterations. Give BLEU score for each iteration.

Note that you can display additional decoder information using `verbose` option. There are several levels with increasing amount of details. e.g. Level 2: `--decoder-flags -v=2`

You can also specify the directory for MERT intermediate files using `--working-dir` option.

For additional details refer to the tutorial on MOSES website.

3) Run MERT on a development set with pruning

Now you will try some additional parameters that will help speed up the decoder.

Translation Table Size (`ttable-limit`):

This is the number of translation per source phrase. The default is 20.

Beam Size (`s`):

The stack size for best partial hypotheses. The default is 100.

Experiment with different values for translation table size (10,3,1), keeping the default stack size, and with different beam sizes (1000,10,1), keeping the default number of translations per source phrase. For each of these parameter settings run the MER training as in Task 2.

You can set these parameters either under `--decoder-flags` or in the configuration file `moses.ini`.

For each configuration, provide for the final iteration

- the resulting BLEU score, and
- the following decoder statistics (average over the dev set):
 - i) Total Number of hypotheses generated
 - ii) Number of hypotheses recombined
 - iii) Number of hypotheses pruned
 - iv) Number of hypotheses discarded early

Interpret these numbers, i.e. are there any interesting differences for the different parameter settings.

4) Run MERT with different initial parameters

Run the MERT optimization with at least 3 other configurations (i.e. different initial values for `weight-d`, `weight-l`, `weight-t` and `weight-w`) and report the BLEU scores for the final iteration. Does optimization end up in the same final configuration?