INF5820, Assignment 3: Machine Translation

Third part, fall 2012

You will now build a small SMT system step by step using Moses. We will roughly be following the recipe at

`http://www.statmt.org/moses/?n=Moses.Baseline`

with some modifications and extensions.

First of all you may choose your language pair. One language will be English. The other will be one of

- Czech
- French
- German
- Spanish

An advice is to choose the language with which you are most familiar (if any). You will find the language resources for this part already downloaded to /projects/nlp/corpus/ .

3 Corpus preparation

Find the section at the web page with the heading "Corpus Preparation" and start at the paragraph "To prepare the data...". The easiest for you is to make a directory "corpus" in your own area where you store the files you create when you refine the corpora. The programs are also downloaded and installed at /projects/nlp/ You may follow the recipe on the web page but the paths must be modified, e.g.,

```
/projects/nlp/mosesdecoder/scripts/tokenizer/tokenizer.perl -l es \
< /projects/nlp/corpus/training/news-commentary-v7.es-en.es  \
> ~/corpus/news-commentary-v7.es-en.tok.es
```

Observe that the tokenizer script takes a language argument "-l" which must be edited properly.

For the next command, the input file is now to be found in your own folder and the modified command becomes

```
/projects/nlp/mosesdecoder/scripts/recaser/train-truecaser.perl \
--model ~/corpus/truecase-model.es \
--corpus ~/corpus/news-commentary-v7.es-en.tok.es
```

etc.

4 Language model

You may here follow the recipe at the web page. With more time we would have stopped and looked what is meant by "improved Kneser-Ney" and how that compares to other smoothing techniques, but to manage the essentials we just follow the web page recipe. Make a directory "lm". We have already installed the irstlm program. You may download it, but you may also run it where it is, e.g.,

```
/projects/nlp/irstlm/bin/add-start-end.sh ....
```

You should also set the path according to where the irstlm program is.

```
export IRSTLM=/projects/nlp/irstlm
```

5 Training

5.1 Size

We will inspect the effects of the size of the training texts: How large amounts of training text is needed to build an SMT system? What is the effect of the size of the training corpora? The web page claims that "for this system I'mm going to use a small (only 130,000 sentences!) data set". How small is that? To how many ordinary printed book pages does this correspond? Use the "wc" command and make your own assumptions explicit to answer this. To how many book pages does the Europarl corpus that you also find in /projects/nlp/corpus correspond? What is the average sentence length in terms of words for your training corpus for the two languages?

5.2 The training

We are going to run two series of experiments, one on a smaller text and one on the whole newstest2011. You should extract a subtext of your training corpus consisting of 15000 sentences. Call it "small.en" and "small.es" (or "small.fr", etc.). You will get an individual number n and you should extract from sentence number n to sentence number $n + 15000$. This may be done on the model of

```
~/corpus $ head -n10 newstest2011.true.en | tail -n5 > small.en
```

which extracts sentences $6 - 10$.

We need two working directories one for each set of experiments. Call them "small_work" and "working". To train on the whole corpus, you may

copy the commands from the web page verbatim except for the path to mosesdecoder, and the name on the foreign language. You must also include a part which tells where GIZA++ was installed

```
-external-bin-dir /projects/nlp/external-bin-dir
```

To train on the smaller corpus, you go to "small_work". In this case you must also change the name on the corpus which is argument to "-corpus" to "~/corpus/small". The language model should be the same for the two experiments.

Training on the full corpus takes a couple of hours, training on small takes more like 15 minutes. Hence, you might choose to train on small first to see that it works and leave the full training to run while you are asleep or at a lecture.

6 Testing - first round

We skip the tuning for now. You may then start the system by

```
/projects/nlp/mosesdecoder/bin/moses \
-f ~/small_work/train/model/moses.ini
```

and similarly for the large system. It takes a couple of minutes to start up; then you can translate sentences interactively. What do you think of the quality?

The web page also describes how you may binarise the model to get a faster start-up. This step is optional. The start-up time does not matter that much when we translate a text file.

Move to "At this stage, your probably wondering how good" at the web page. Prepare the test corpus as described. You don't have to run the filtering process.

Translate the test corpus, where you find the "moses.ini" the same place as before, "~/small_work/train/model/moses.ini", and similarly for the full system.

Then calculate the BLEU-score for the two translations.

7 Tuning

7.1 Run the tuner

We are now entering the world of magic, and see what tuning may do for our two systems. Scroll up to the section Tuning on the web page. Prepare the

dev corpus. You may then go to the two directories small_work and working and run the tuning command in turn. Adjust the path to mosesdecoder properly. Also, beware that the tuning is really slow. You should definitely include the

```
--decoder-flags="-threads 4"
```

But even then, the tuning takes several hours. For "small", it took between 2.5 and 3.5 hours for me, for the full corpus, it took more than 7 hours, i.e., you need a night.

7.2 Test the tuning

Now test the tuned systems on the same test set as the systems before tuning. The path to "moses.ini" will now be "~/small_work/mert.work/moses.ini". Have the numbers become better?

What to deliver?

- Information: Which language did you choose?
- Answers to the questions in subsection 5.1 Size.
- The full output from the BLEU test of the 4 systems (sections 6 and subsection 7.2)
- The directories and files which are produced—i.e., corpus, lm, small_work, working with their content—should *not* ne handed in, but be placed somewhere in your area where it is available for reading by the instructor. You have to set proper protections and deliver a path to where to find them.

End of assignment