Lab 6: Quality automatic evaluation

Instructions to follow the lab n.6 GIAN course about MT Status: 13/12/2016 BP V0.01

Preliminaries

1. Your host compute contains a "SMT" virtual machine provided
2. Open virtual box (on Ubuntu, click on "search", type virtual, open it)
3. The virtual machine will appear, select the machine, click "start"

The username is "smt" and password is the same

The corpus (reminder)

The "indic languages" corpus was downloaded from http://homepages.inf.ed.ac.uk/miles/data/indic/. It contains a set of sentences extracted from Wikipedia that were translated by 4 different persons using mechanical turk. Each English sentence may have 4 different Hindi versions. Each entry in the testset and devset has one Hindi version but 4 different English translations (hence the unique file test.hi and the 4 files test.en.0…test.4.en)

Launch a BLEU evaluation on the trained model

Open a terminal, go into the "trainMoses" directory

```
cd ~/parallel-corpora/hi-en/trainMoses/
```

If you have not done it before, launch the translation of the Hindi testset and save the result in a file call "output.en"

```
cat test.hi |~/mosesdecoder/bin/moses -threads 3 -f model/moses.ini > output.en
```

(note the option "-threads 3" that launches Moses in parallel on 3 cores)

Check that the input file and the output file have exactly the same number of lines

Now launch a BLEU evaluation using the multi-bleu script provided with Moses.

Note that the usage of multi-bleu is (as shown when you launch it without any parameters):

```
usage: multi-bleu.pl [-lc] reference < hypothesis
Reads the references from reference or reference0, reference1, ...
```

Launch a first evaluation, remember that we compare testset.en to the output.en:

```
~/mosesdecoder/scripts/generic/multi-bleu.perl test.en.0 < output.en
```

or

```
cat output.en |  ~/mosesdecoder/scripts/generic/multi-bleu.perl test.en.0
```

➔This will show you the BLEU score when comparing to a single reference test.en.0

Try it with each of the 3 other reference translation. Which one of the “mechanical turc” did translations that are closer to the machine output?

Remember that BLEU can take more than one reference, now please launch it with the 4 references:

```
~/mosesdecoder/scripts/generic/multi-bleu.perl test.en.0 test.en.1 test.en.2 test.en.3 test.en.4 < output.en
```

Do a similar evaluation on the devset

Create a “pseudo-wrong” testset by taking the last 1000 lines of the trainset

Translate it using the model and compute the BLEU score

- ⇨ Observe that the BLEU is higher, that’s a normal behavior: a MT is always better on something it was trained on

TER: Translation error rate

Now try to compute the translation error rate:

Usage: perl ~/Icon2016/lab/ter.pl translation reference

e.g

```
perl ~/Icon2016/labs/ter.pl  test.en.0 output.en
```

(note that the TER script does not allow you to compute comparison with multiple references)

Do similar experiments: compare each translator’s version, then compare 2 translators.

Use the TER how “distant” are the 4 translations done by the “mechanical turcs”

Similarly use the BLEU score to compute a comparison between the 4 translators

Sentence Bleu

As you may know BLEU can easily be 0 for short sentences, let’s try to look at the quality of the testset sentence by sentence using the “sentence_bleu” script provided with Moses

Usage is the same as “multi-bleu” only that it will now display a smoothed-BLEU score for each line

Launch a sentence-bleu evaluation and use Unix command to display a tab-separated file containing:

```
Smoothed-BLEU<tab>English<tab>Hindi
```

Comparison of two models

Compute the improvement of the scores (both BLEU and TER) on the Malayalam training, try to launch the Malayalam training after adding the “dictionary” data (see assignment of yesterday, last part).

And compute the BLEU to observe if adding this new data made a difference