A guide to build Neural MT system @ Fondazione Bruno Kessler

Rajen Chatterjee
chatterjee@fbk.eu

18th July, 2016

1 Introduction

This documentation[1] contains a step-by-step procedure to build a neural machine translation (NMT) system using the Nematus toolkit[2], which is based on the code in dl4mt-tutorial by Kyunghyun Cho et al..[3] An optional preprocessing step for word segmentation based on byte-pair encoding (bpe) can be performed using the Subword[4] (see Section 5 to enable or disable this step). The Nematus toolkit and the Subword are installed respectively at `/hltsrv1/software/nematus-master/` and `/hltsrv1/software/subword`.

2 Training from scratch

1. Create a data directory (`/home/data`) with the following files:
   (a) train.*src*
   (b) train.*trg*
   (c) dev.*src*
   (d) dev.*trg*

   where *src* and *trg* can be any language code (eg. en, de, fr,...). The data must be already pre-processed (eg. lower-cased, normalized, tokenized, true-cased) based on user requirements. Only the subword (bpe) preprocessing is handled in this framework (see the config file).

2. Copy the configuration file from `/hltsrv1/software/nematus-master/config.cfg` to `/home/data`

[1] This is an on-going documentation and will be updated frequently. Any feedback/comments/suggestions are more than welcome
[2] https://github.com/rsennrich/nematus
[3] https://github.com/nyu-dl/dl4mt-tutorial
[4] https://github.com/rsennrich/subword-nmt

3. Edit the config file (`/home/data/config.cfg`) to set various paths and to define network architecture.

4. Login to a gpu machine: (if your system runs as expected then you can submit the job with qsub)
   `qlogin -q gpgpu.q -l mf=500G,gpu=1`

   - `mf=500G` means you ask for 500GB memory. It is observed that the system crashes with smaller value for memory, this value (500GB) is reliable and also been authorized by the cluster head.
   - `gpu=1` means you ask for 1 gpu core (out of 4 core in one machine). Currently, this framework support only one core so asking for more core will not be useful.

5. Start training
   `/hltsrv1/software/nematus-master/train.sh $path-to-config.cfg $gpuID`
   (where gpuID: gpu0, gpu1,gpu2,gpu3)
   to check which gpu is free run the following command in a gpu machine
   `> nvidia-smi`
   this will list all the available gpu and the ones that are currently occupied by other processes

6. (Optional) To submit a job with qsub use the following command:
   qsub `/hltsrv1/software/nematus-master/train.sh $path-to-config.cfg $gpuID`

7. To stop the training, press ‘‘`CTRL + C`’’ or ‘‘`kill processID`’’

3 Training from an existing model

1. Set the flag “`reload_=True`” in the configuration file (`/home/data/config.cfg`)

2. Follow steps 4 to 7 mentioned above (i.e. qlogin and then start training)

4 Decoding test set

Run the following command:
`/hltsrv1/software/nematus-master/translate.sh $path-to-model.npz $test.src $reference $gpuID`

1. The decoder uses a beam search size of 12

2. This command will generate

   (a) hypothesis file (the MT output) test.src.ouput

(b) word alignment probability file test.src.output.align

(c) clean hypothesis file (the MT output is post-processed to combine the subwords) test.src.output.postprocessed

3. BLEU score is computed between the reference and test.src.output.postprocessed

To change the parameters of the decoding, copy the script `/hltsrv1/software/nematus-master/translate.sh` to `/home/data` then change the settings as required and run the script.

5 Configuration

Some of the useful parameters to tweak are listed in Table 1

Parameter	Description	Recommended value
data_dir	the directory containing the training and development corpus	-
work_dir	the directory where the output files will be saved	-
src	file extension of the source corpus (eg. en, de, fr)	-
trg	file extension of the target corpus (eg. en, de, fr)	-
bpe_operation_src	number of codes to learn from the source corpus	40000
bpe_operation_trg	number of codes to learn from the target corpus	40000
apply_bpe	subword pre-processing to be performed	1
n_words_src	size of the source vocabulary	40000
n_words	size of the target vocabulary	40000
maxlen	sentences with length greater than maximum length will be discarded in training	50
dim_word	size of word embedding	620
dim	size of hidden unit	620
batch_size	number of samples used to learn parameters for each update	100
valid_batch_size	number of samples used from the development set	100
reload_	whether to start the training from previously saved model	True
saveFreq	number of updates after which the model will be saved	10000
overwrite	should the previously saved model be overwritten	False
validFreq	number of updates after which the development set will be used to evaluate systems' performance	10000

Table 1: Parameters in the config.cfg file