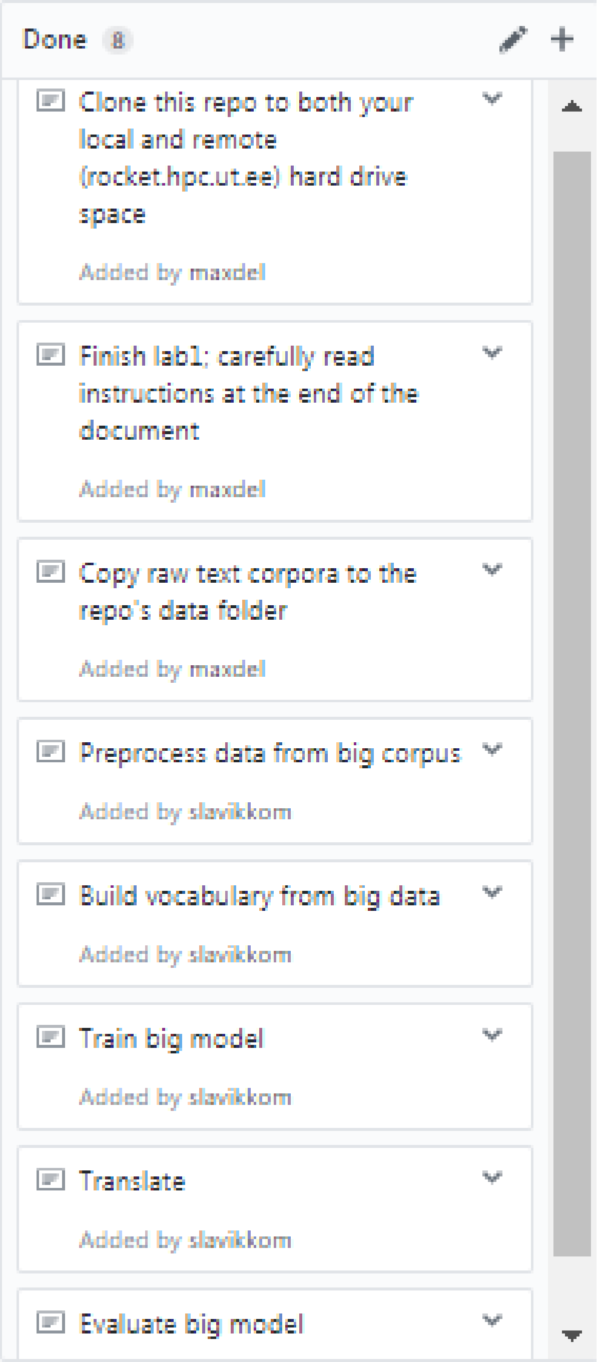
First milestone report

“Researchers” team:

https://github.com/dil-delada

https://github.com/slavikkom

Our project board



Done 8

Clone this repo to both your local and remote (rocket.hpc.ut.ee) hard drive space

Added by maxdel

Finish lab1; carefully read instructions at the end of the document

Added by maxdel

Copy raw text corpora to the repo's data folder

Added by maxdel

Preprocess data from big corpus

Added by slavikkom

Build vocabulary from big data

Added by slavikkom

Train big model

Added by slavikkom

Translate

Added by slavikkom

Evaluate big model

This milestone tasks description

1. Corpus Preparation

Starting from raw data, we applied following preprocessing steps:

a. Corpora concatenating (https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/preprocesing.sh):
we got one big parallel text corpus of **4478200** lines;
b. Data shuffling (https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/preprocesing.sh**)**:
to feed sentences to NMT system later in the random order;
c. Data splitting (https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/preprocesing.sh):
4378200 training examples, **50000** test examples, and **50000** development examples;
d. Tokenizing all the sets (https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/preprocesing.sh);
e. True casing all the sets(https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/preprocesing.sh);
f. Filtering out empty and strange sentence pairs from the training set. (https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/preprocesing.sh);
g. BPE (https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/build_vocabulary.sh):
we used joint vocabulary of size 30000.

Note: We used Moses scripts to do basic preprocessing, and BPE for the subword segmentation.

2. Model Training

We used 1 Tesla P100 GPU Machine provided by HPC center of the University of Tartu to train our model with vocabulary of size **30000**. The model we trained is the default OpenNMT-py model, which consists of a 2-layer LSTM with 500 hidden units on both the encoder/decoder.

We had trained our best model for ~**3** days, **13** epochs. Development set perplexity was **2.86**. We **waited to the end** to stop the training process.

You can find the script we used to run training here (https://github.com/mt2017-tartu-shared-task/nmt-system-C/blob/master/scripts/train_model.sh).

3. Translating and Evaluating Results

We performed an inference and got unpostprocessed English hyps file.

We used this file, processed reference file, and BLEU metric to evaluate the translation performance of our model, and got **37.38** points.