MTAT.06.055 Machine Translation

In-class Machine Translation Shared Task 2017

Olha Kaminska and Viacheslav Komisarenko

UNIVERSITY OF TARTU
Institute of Computer Science

Task

There is given an Estonian - English corpora of approximately **19000000** sentence pair. We needed to train neural machine translation model that translates Estonian texts in the best way. Evaluation of results is based on **BLEU** score on test set provided in the end of competition. In the middle of competition, we also received small texts for validation purposes.

Plan of project

On the Fig. 1 illustrated all main steps that we performed during task solving. Further they described precisely.

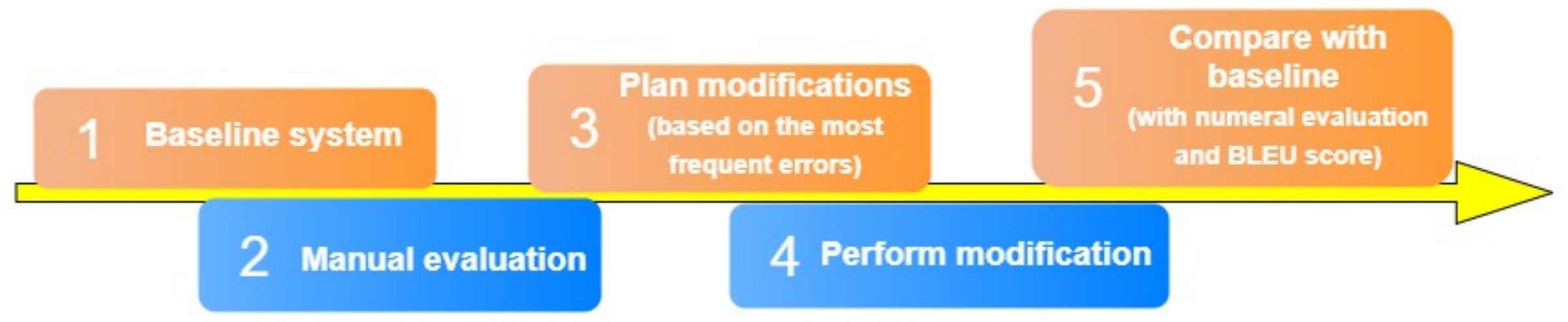
1 Baseline system
2 Manual evaluation
3 Plan modifications (based on the most frequent errors)
4 Perform modification
5 Compare with baseline (with numeral evaluation and BLEU score)

Figure 1: General plan.

Data preparation

This tutorial assumes that we work with translation direction **Estonian -> English**.

In practice, we begin with some raw data in parallel text format (parallel corpus), or even with several datasets like this (parallel corpora) that come from different sources. In order to prepare data for NMT training, we did the following **steps**:

1. Corpora concatenating: we got one big parallel text corpus of 19051439 lines;
2. Data shuffling: to feed sentences to NMT system later in the random order;
3. Data splitting: 18951439 training examples, 50000 test examples, and 50000 development examples;
4. Tokenizing all the sets;
5. True casing all the sets;
6. Filtering out empty and strange sentence pairs from the training set;
7. BPE: we used joint vocabulary of size 30000.

In order to do steps 2-4 we used **Moses scripts** and BPE for the subword segmentation.

Baseline 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 OpenNMT-py model with default parameters, which consists of a 2-layer LSTM with 500 hidden units on both the encoder/decoder.

We had trained our best model for **5 days, 13 epochs**. Development set **perplexity was 4.05**. We waited to the end to stop the training process.

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 **31.5 points**.

Manual evaluation

We manually analyzed **40 baseline translations**.

Our main observation was that the most of the errors in our results were caused by fact, that machine used synonyms, which have a little **different meaning** and it changes sense of the sentences. Several times some words were missed or was used incorrect tense. Take a look at our the motivating example produced by baseline system:

Example 1.
Estonian: Meie oskus aidata luua vrtusphiseid teenuseid vib anda tulemuseks kiirema vrtusteni judmise.
Human English Translation: Our ability to help create asset-based services can result in faster time to value.
Machine English Translation: Our ability to help create valuable services can result in faster values.
Example 2.
Estonian: Te kontrollite kogu kasutuskogemust.
Human English Translation: You control everything about the experience.
Machine English Translation: You're checking all the experience.

Planned modifications. Based on manual evaluation of machine translation quality, we conclude that for problems of inappropriate synonyms it is good idea to try ensemble decoding and optimal choice of beam size.

Final system

In order to address translation issues found after our manual evaluation we train model with **Nematus framework** (with default parameters) and translate with ensemble decoding (using three best models).

For translation we also tune best value of beam size.

The trained system gave us 16.8 BLEU points on the shared dev set that means decrease over the baseline.

Manual evaluation of final system

Generally speaking, results are slighly worse: grammar is approximately on same level, but use of incorrect words is really often. Let's now look at how does our Motivating example looks like with our final system:

Example 1.
Estonian: Meie oskus aidata luua vrtusphiseid teenuseid vib anda tulemuseks kiirema vrtusteni judmise.
Human English Translation: Our ability to help create asset-based services can result in faster time to value.
Machine English Translation: Our ability to help create value-based services can lead to faster values.
Example 2.
Estonian: Te kontrollite kogu kasutuskogemust.
Human English Translation: You control everything about the experience.
Machine English Translation: You'll check all the use.

As a result you can see that baseline translation is better. Final system translation is less fluent and often use incorrect words (but overall structure of sentence looks fine).

Beam size tuning

After training Nematus model and applying ensemble decoding, we tried to find best beam size.

We iterate through grid from 1 to 21 and calculate BLEU score on validation data.

As we can see on Fig.2, graph has local maximum near 11. However, after 17 BLEU slightly higher, and stabilize with next growth of beam size.

As best beam size value was chosen 18.

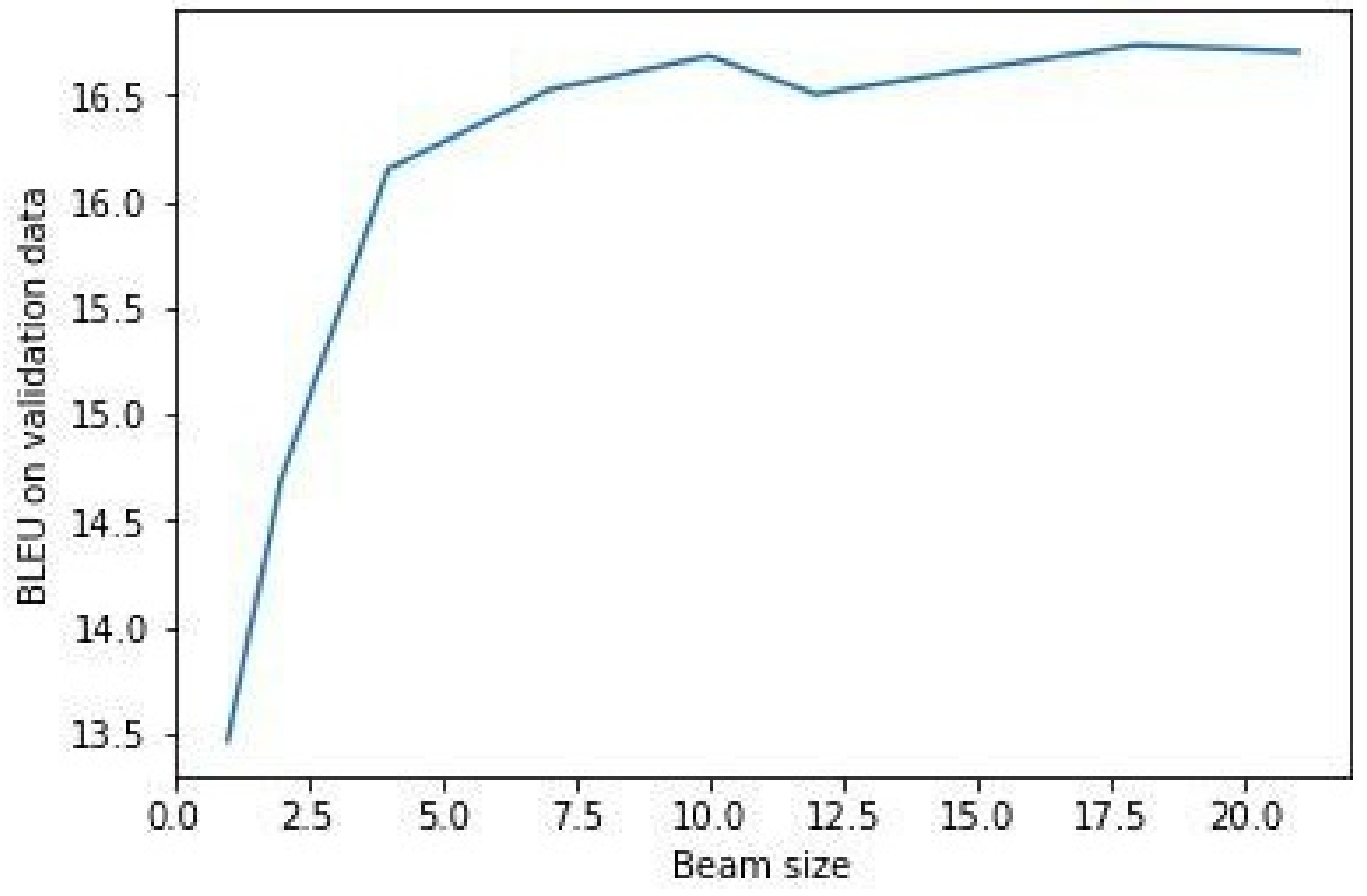

Figure 2: BLEU score for different beam size.

What we also tried or wanted to try

In the beginning, we mistakenly start training baseline model not at all texts but on some subset of size **4500000** sentence pair.

As a result, we surprisingly receive high BLEU score of **37.38** on local test set. However, terrible score on provided validation data shows need of retraining.

After retraining, we also tried to continue training baseline system one week more. After reaching 20 epochs (in baseline was 13 epochs) results converge, so, there were no need to train default system more. However, it does not influence results a lot.

Training with Nematus framework continued one week. We wanted to train model for more time (one-two week more), because we found Google Cloud GPU (one Tesla K80) quite late and one week was not enough.

Contact

Github repository of our project: github.com/mt2017-tartu-shared-task/nmt-system-C

Our Github profiles:

- github.com/slavikkom
- github.com/dil-delada

Advisors: Mark Fishel , Maxym Del