2aii) Average of 5 BLEU scores for each dataset

Dataset	Average BLEU scores over 5 Folds
cs	0.006926144028
ru	0.006475868922
fr	0.007184015044
de	0.008783822406

2aiii) **Question 3:**

-30000 epochs

-Greedy Decoding

Dataset	BLEU Scores for Question3
cs	0.0014870223554579816
ru	0.0027476273994840273
fr	0.0034782421869055864
de	0.004020187179396985

Question 6:

-30000 epochs

-Beam Search Decoding, with beam size of 2

Dataset	Without max_length With normalization	Without max_length Without normalization	with max_length With normalization	with max_length Without normalization
cs	0.010070669217292215	0.0017089251013881822	0.0029892619122392863	0.0008778783868552329
ru	0.006179483317263897	0.0016558623310648627	0.0036022767900668897	0.0015078081467405809
fr	0.0038480025206779622	0.0016839998675670848	0.0036247172163981043	0.0021027743867321974
de	0.003763959965891697	0.0024950939487778175	0.004129606688140779	0.0024568692307975104

(Results in red indicates worse performance compared to the implementation in Question 3)

For cs data set: With max_length and without normalization, beam search is less effective than the implementation in Question 3. One reason may be the lack of normalization since there are varying length of sentences in the dataset. Longer hypotheses tend to have lower scores; therefore, normalization is used to fix this issue.

For ru data set: Without normalization, beam search is less effective than the implementation in Question 3. Longer hypotheses tend to have lower scores; therefore, normalization is used to fix this issue

For fr data set: Without normalization, beam search is less effective than the implementation in Question 3. Longer hypotheses tend to have lower scores; therefore, normalization is used to fix this issue

For de data set: Without normalization, beam search is less effective than the implementation in Question 3. Longer hypotheses tend to have lower scores; therefore, normalization is used to fix this issue.
With normalization, without max_length, beam search is less effective than the implementation in Question 3. Since longer hypotheses tend to have lower scores, without a max length, the decoding with result with longer hypotheses.

2bi)

The attention decoder of Tutorial 6 uses GRU while the attention decoder of the lecture uses RNN. GRU is different from RNN in that GRU has 2 gates inside it. One reset gate and one update gate. Both gates have its own weight and biases. The reset gate determines whether the previous cell state is important or not. The update gate decides if the cell state should be updated with the new state.

2bii)

-30000 epochs

-Greedy Decoding

Dataset	BLEU scores using RNN
cs	0.0021230250502493045
ru	0.002402296081954876
fr	0.0006301196309412273
de	0.01654010736034334

For the cs and de datasets, RNN performed better. For the ru and fr datasets, GRU performed better. The difference in performance may be due to the difference in language structures. Some language structure performs better with different neural network architecture.

2biii)

1)

Multiplicative attention (class AttnDecoderRNN_mul):

```
# Calculating Alignment Scores using Multiplicative attention
out = self.fc(rnn_out)
alignment_scores =  torch.bmm(out.view(1,1,500),encoder_outputs.unsqueeze(0))
```

2)

Additive attention (class AttnDecoderRNN_add):

```
# Calculating Alignment Scores using Additive attention
x = torch.tanh(self.fc_hidden(hidden[0])+self.fc_encoder(encoder_outputs))
```

4)

	BLEU Score
Question 3 of Tutorial 6	0.1638904170559944

The evaluation results for Question 3 of Tutorial 6 is much better than those of 2a and 2bii. This is because the dataset being used for Question 3 of Tutorial 6 is simpler since the sentences have been filtered down to a maximum of 10 words per sentence. The translation task is therefore simpler. In contrast to the 4 datasets used in this assignment, where there is no maximum number of words per sentence used. Furthermore, the Russian dataset uses the Russian alphabet, which adds another layer of complexity during translation.