Reports on Neural Machine Translation

Github link: https://github.com/TheUsernameIsNotTaken/NMT_SentencePiece

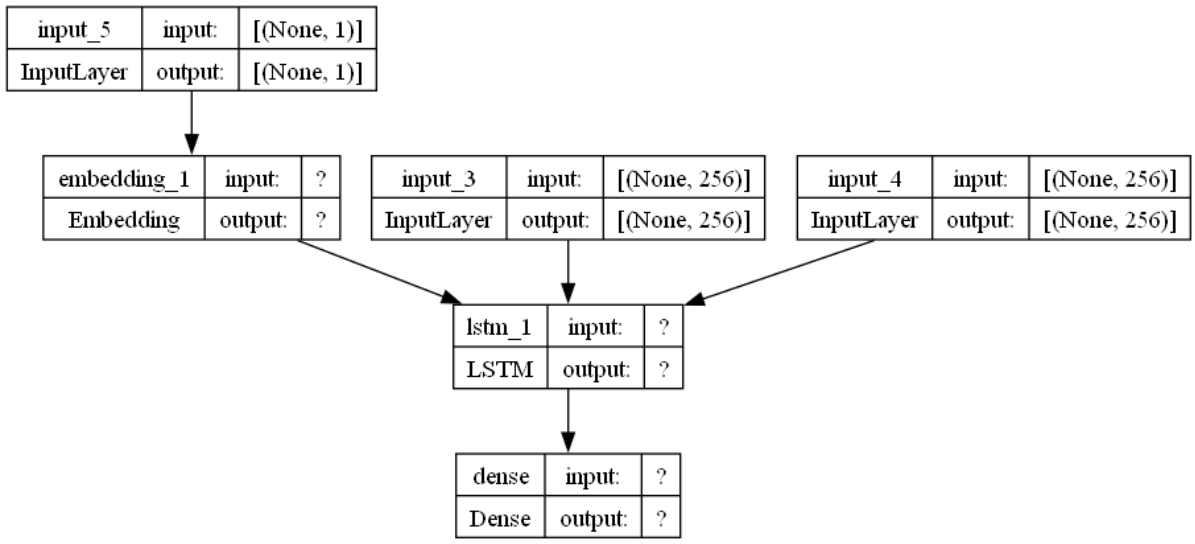
Trained models

I overall trained 3 different models, in both English to Hungarian, and Hungarian to English. I only did score-validation for the English to Hungarian one, but I think the language pair, ont he other way around, still I will mention here, that the overall Fluency increase can be also seen on this translation direction, and it also increases the adequacy a little, but at least it is clearly seeable it does not make it worse.

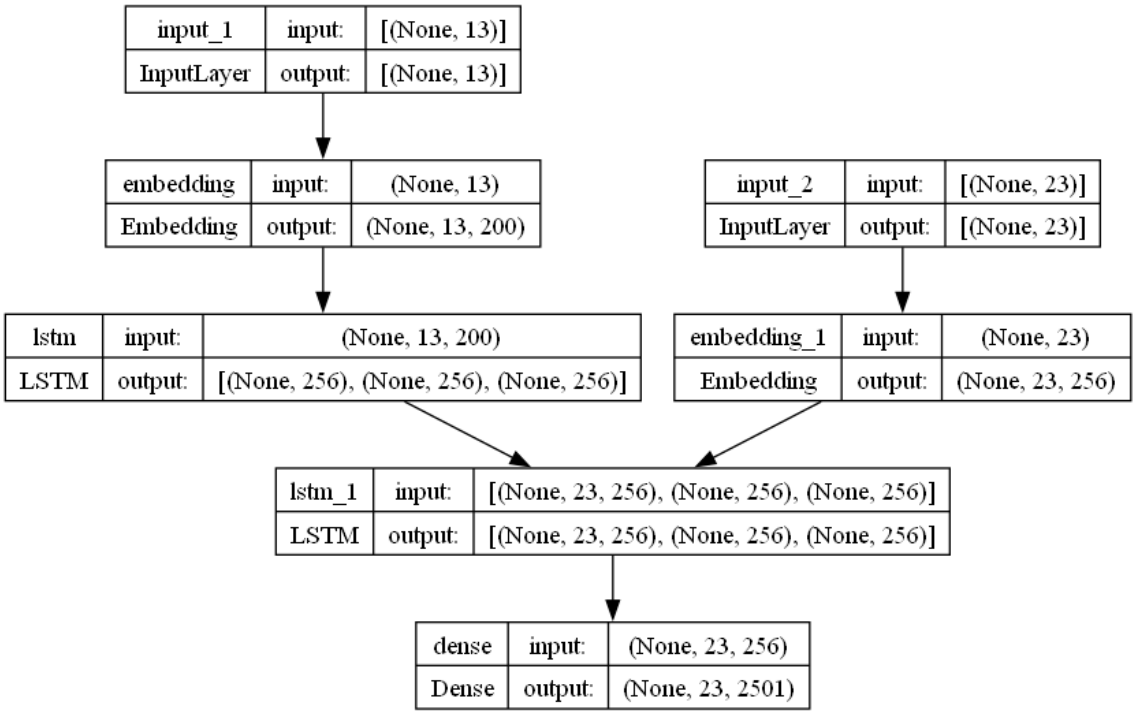
The three models are the following: In model 1 I use words as tokens. Then on the other models, I changed up the tokenizer to SentencePiece. On the second I trained the SentencePiece tokenization only on the training dataset, and on the third one, I trained it on a bigger dataset. It was a book, titled „Az aranyember" or „Timar's two words", by Mór Jókai. Because the text is from the same book, providing us a nice bilingual dataset.

I generated the same models for Hungarian-to-English translation too, and they are similar to these.

All of the models have the following pre-setup look.



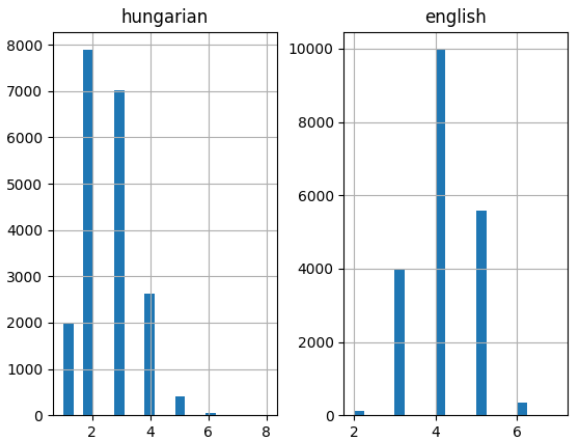
After training they look like this, the only difference is the dimensions of the input and output, because that depends on the tokenized sentences maximum token numbers.



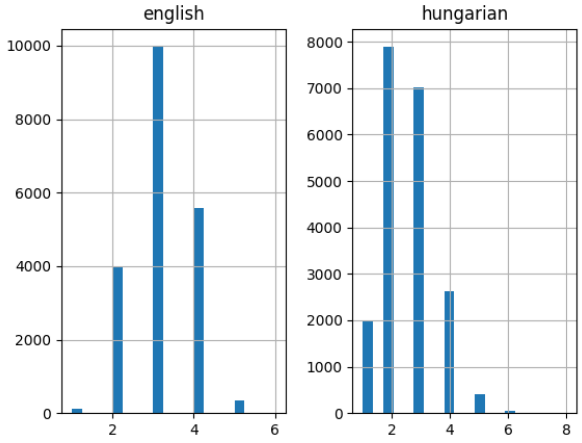
I attached the codes (v3 is model 1, and v4_tk is model 2, while v4_th_sbs is model 3), the SentencePiece vocabularies and models (where "_sbs" is attached at the end, it means that it is the „simple data set" one, meaning it only used the model training data to train up the tokenizer) and the trained NMT models. Each of them can be found in the github repo.

I also attached a folder named output-commands, where the outputs of the training and tests can be found. The training datasets:

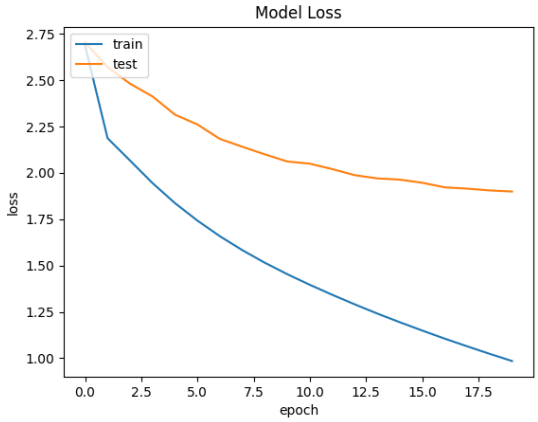
Hun to Eng:



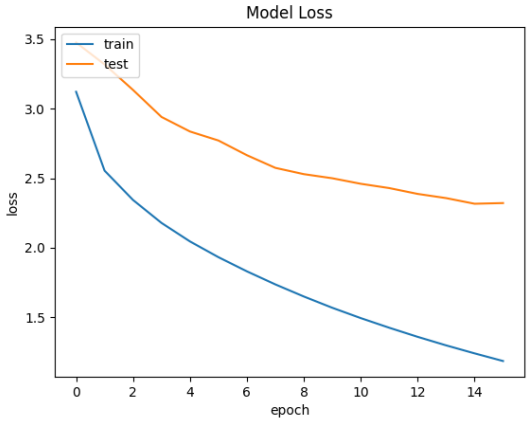
Eng to Hun:



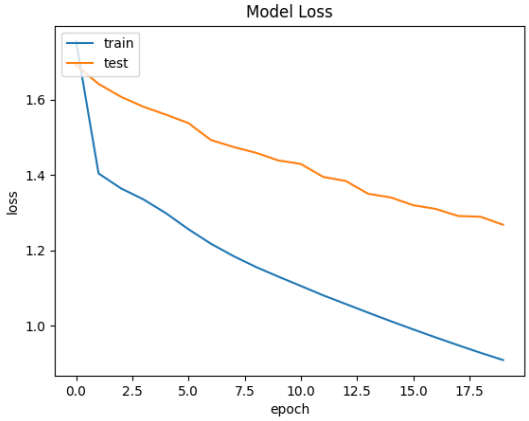
Model 1 eng to hun loss on epochs:



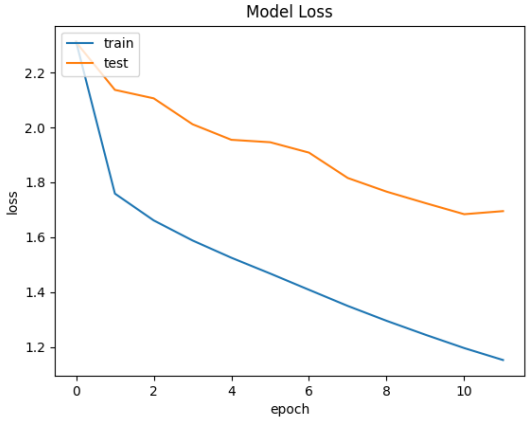
Model 1 hun to eng losson epochs:



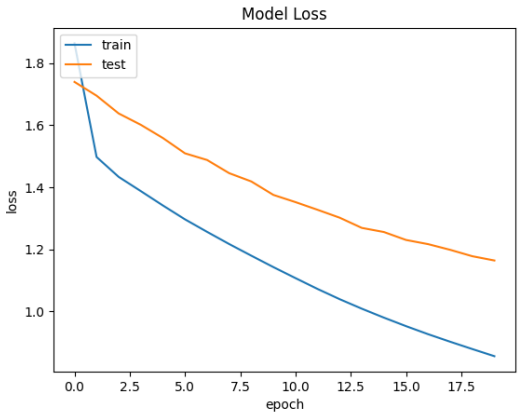
Model 2 eng to hun loss on epochs:



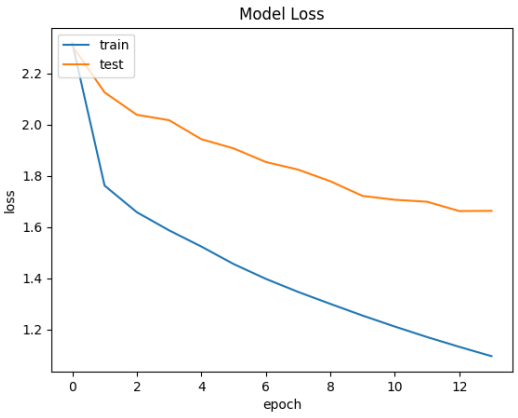
Model 2 hun to eng losson epochs:



Model 3 eng to hun loss on epochs:



Model 3 hun to eng losson epochs:



As I can see from these, the loss is decreases more quickly, and the models reaches convergence sooner when the target language is English. (Probably because that side has usually fewer tokens in one sentence in any model.)

Validated models

I did a human validation with a short pool of sentences (30). I pointed their adequacy and Fluency by the following table (what was part of the study material)

Adequacy	
5	all meaning
4	most meaning
3	much meaning
2	little meaning
1	none

Fluency	
5	flawless English
4	good English
3	non-native English
2	disfluent English
1	incomprehensible

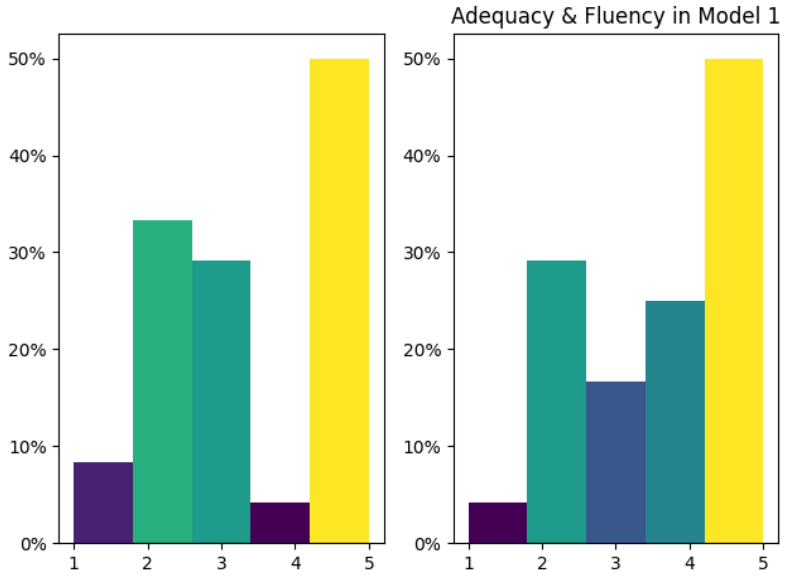
The three English to Hungarian (ENG → HUN) models. Their metrics are the following:

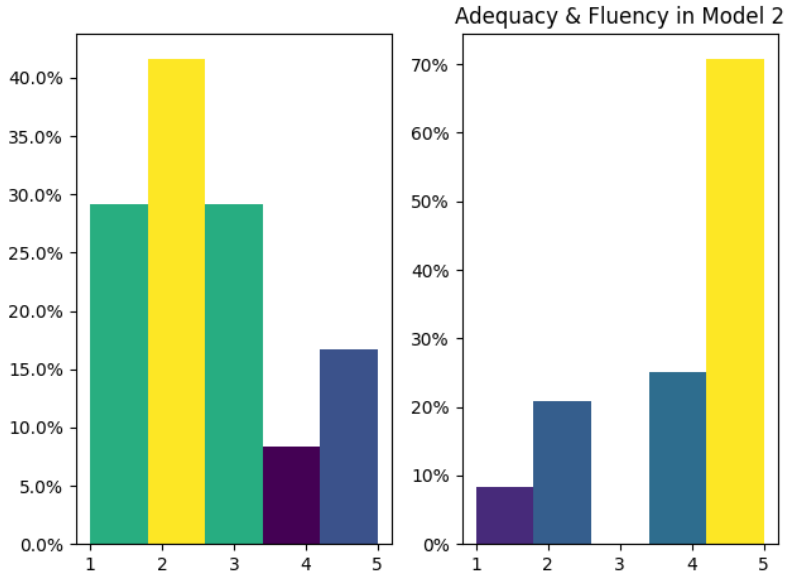
Translation Languages			ENG --> HUN		
Tokens	Tokenizer type		Word-based	SentencePiece	
	Tokenizer training dataset		Only training data	Trained on training dataset	Trained on big dataset (book)
Metrics (last round)	Accuracy	Training accuracy	0.8257	0.8300	0.8558
		Validation accuracy	0.7820	0.7851	1.1642
	Loss	Training loss	0.8558	0.9094	0.8257
		Validation loss	1.1642	1.2684	0.7820

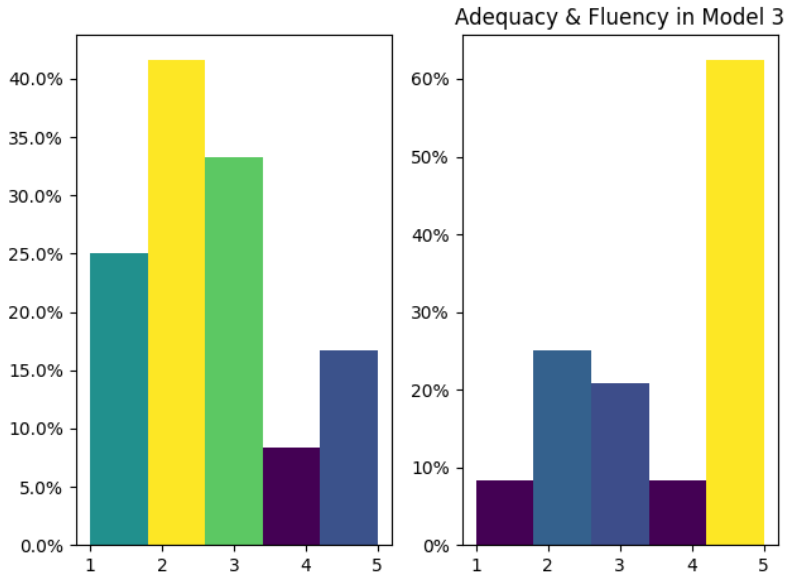
Neither of the generated models are terrible per say, they can usually create understandable sentences, but the meanings can be hit-or-miss.

I see that the reults might be biased because I used sentences what are part of the training set too.

The whole point table can be found in the „ENG_HUN_validation.xlsx” file, and the histograms of the points are included in the following page. The model number (1, 2 or 3) means the same, what I mentioned before, and what is shown in the validation diagram. The histograms was generated by the „gen_histograms.py” python script. The histograms (all of them are in the next page):


Adequacy & Fluency in Model 1
50%
40%
30%
20%
10%
0%
1
2
3
4
5


Adequacy & Fluency in Model 2
40.0%
35.0%
30.0%
25.0%
20.0%
15.0%
10.0%
5.0%
0.0%
70%
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5


Adequacy & Fluency in Model 3
40.0%
35.0%
30.0%
25.0%
20.0%
15.0%
10.0%
5.0%
0.0%
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5