Anni Moisala

015173294

LDA-T306

FINAL PROJECT

Language pair and data

Data selection

I chose German-English as my language pair for this project. Although I don’t speak German, I wanted to try some other language than my mother tongue Finnish. For my data source I chose WikiMatrix, which is a corpus compiled from Wikipedia articles (Schwenk et al. 2019). Wikipedia covers many languages and topics and is publicly available, so it seems like a good source for a parallel corpus. Possible problems with the data are that it is not very clean, and it is not clear what it contains.

Preprocessing

First I filtered out long sentences, which I defined as sentences exceeding 100 words. Next I used a language ID filter, since the data had quite many sentences in a wrong language, mainly German in the English data. After the filtering I inspected 1000 lines from both languages, and I could not spot any sentences in wrong languages. Then I took the first 5000 lines for testing, the 5000 lines after that for validation and the next 1M lines for training the baseline model.

Phase 1: Baseline model

Attention

In the previous assignment I got the best results with general attention with the input feed off, so I kept these settings. I evaluated the model and got a BLEU score of 42.1 and chrF2 63.5 using my own test set.

BPE units

I decided to use subword splitting for this model and kept the amount of BPE units the same as in the previous assignment, which is 32k.

Layers

Using a model based on the previous assignment and settings described above, I experimented with the amount of layers. As in the assignment, increasing the layers lowered the evaluation scores, due to the small amount of training data. Therefore, I kept the amount of layers as 2.

RNN and word vec size

I started with these parameters set at 512, then I increased them to 612 and found that it improved the BLEU score by 0.3 and chrF2 0.7 using my own test set. I set these values at 612.

Training steps

First I set my training steps to 400 000 since the models were not training for very long, between 4 and 6 hours. However, I found that the accuracy did not improve after 200 000 training steps, so I continued with that.

Dropout

After finding optimal settings for parameters I had used in previous assignments, I set the dropout to 0.1. That increased the BLEU score by 1.8 and chrF2 by 1.4 using my own test set. I kept this setting for the first model.

Encoder type

Next I tried setting the encoder type to brnn to use a bidirectional encoder and evaluated the model compared to the previous one and that increased the BLEU score by 1.6 and chrF2 by 1.3. I kept this encoder type for the first model.

Adam

Lastly I set the optimizer to adam with the learning rate at 0.001, this decreased the BLEU score by 1 and chrF2 by 0.8, therefore I decided not to use it in this model.

Final settings

This is how my baseline model configuration file looks like:

```
train_steps: 200000
save_checkpoint_steps: 50000
valid_steps: 25000
global_attention: general
input_feed: 0
dropout: 0.1
world_size: 1
gpu_ranks: [0]
layers: 2
rnn_size: 612
word_vec_size: 612
encoder_type: brnn
```

Testing with my own test set I got a BLEU score of 45.8 and chrF2 score of 66.9. The score improved by 3.7 and 3.4 respectively. With the Tatoeba test set I got a BLEU score of 27.2 and chrF2 score of 47.9. Validation accuracy was 77.

Phase 2: Improvements

Model 2: More training data

For my second model I wanted to try increasing the amount of training data. Since I had plenty of filtered data I increased the amount of sentence pairs to 2M instead of doing back translations.

While using a bigger training data I also wanted to see if adding layers would improve results, so I trained three models, one with 2 layers, second with 4 layers and a third one with 6 layers.

Surprisingly, the models with more training data performed worse than the baseline. To make sure it was not caused by a mistake in the preprocessing, I redid everything and got a slight improvement but still lower scores than the baseline model. Comparing the second attempt with the baseline, the score went down by 3.4 in BLEU and 4.9 in chrF2 testing with my own test set. More layers did not improve the results, as I anticipated with using a bigger training data. Using the Tatoeba test set the BLEU score lowered by 2.3 and chrF22 lowered by 3.1, which is a slightly smaller decrease than with my own test set.

In contrast to my expectations, more data does not always mean better results. One reason for a decrease in performance could be low quality data. Another reason might be that the settings are not optimal for a bigger training data. However, it could be that the model did not train long enough since I used the same amount of training steps. Although the model did reach accuracy of 75.4 at 62 650 training steps, and final accuracy is 75.5. Due to lack of time now I cannot test this hypothesis, since at the time of training I did not consider that and wanted to keep all settings the same to accurately compare results.

Model 3: Transformer architecture

As the third model I used the transformer architecture. I downloaded the example configuration file provided on the course and changed the rnn size and word vec size to match my baseline model and changed the amount of layers to 4. The model was training for 65 hours and reached validation accuracy of 80.8. I tested it and compared to the baseline the BLEU score improved by 6.3 and chrF2 improved by 5.8 with my own test set and with the Tatoeba test set BLEU score improved by 5.7 and chrF22 improved by 6.2, which is quite a big improvement.

This is how my transformer configuration file looks like:

```
encoder_type: transformer
decoder_type: transformer
position_encoding: true
enc_layers: 4
dec_layers: 4
heads: 8
rnn_size: 612
word_vec_size: 612
transformer_ff: 2048
dropout: 0.1
attention_dropout: 0.1

# Optimization
model_dtype: fp32
optim: adam
learning_rate: 2
warmup_steps: 8000
decay_method: noam
adam_beta2: 0.998
max_grad_norm: 0
label_smoothing: 0.1
param_init: 0
param_init_glorot: true
normalization: tokens

# Batching
world_size: 1
gpu_ranks: [0]
batch_type: tokens
batch_size: 4096
valid_batch_size: 4096
max_generator_batches: 2
accum_count: 8
```

Comparing results

Below is a table presenting evaluation scores from all three models.

	my test set	Tatoeba test set
Model 1	BLEU 45.8 chrF2 66.9	BLEU 27.2 chrF2 47.9
Model 2	BLEU 42.4 chrF2 62.0	BLEU 24.9 chrF2 44.8
Model 3	BLEU 52.1 chrF2 72.7	BLEU 32.9 chrF2 54.1

Manual evaluation

Looking at the output files of the Tatoeba test set, all models translate easier sentences really well. With more difficult sentences, a difference in quality corresponding with the scores between the three models can be seen.

For example the sentence “95 years old! God Save the Queen!” is translated like this by the different models:

Model 1: "The Queen Shall the Queen".

Model 2: "The Queen 's Day".

Model 3: “95 years old God save the Queen.”

More differences that I noted between the models is that the model 2, worst performing model, was the only one I noticed producing repetition of words. In addition, model 3, the best performing model is the only one I saw using discourse markers such as “Oh!”, whereas the other models left these out. However, not even the transformer could always produce such words, for example the word “ouch”.

Sources

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong and Paco Guzman, WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia, arXiv, July 11 2019.