Mechanical Translate from English to German

Yibang Hu

Abstract

In this project, I used a pre-training model, which is T5-base model, to do the task: mechanical translation from English to German. In addition, I also trained a random initialization model that didn't use the pre-trained model to do the same translation and compare the BLEU scores not only between those two models, but also with another model which I found based on the same dataset. For the result, I got the BLEU scores of 38.8 on the pre-trained model and 22.6 on the random initialization model.

1 Introduction

In this project, I needed to choose a pre-trained model and learn how to fine-tune that model to make it useable on my downstream task, which is accept English as the input text and output German text.

I tried to use Bert-base model as my transformer encoder or use T5 model to replace my whole model that I used in HW5. I preprocessed the data in another way to match the pre-trained model I was using.

2 Related Works

At first, I plan to use a Bert model as an encoder in transformer that we used in HW5. However, I got a BLEU score below 1 no matter how I change it. I think that probably because my model in HW5 have some problem with a pre-trained model as encoder. Therefore, I changed my decision to use T5-base, a Transformer encoder-decoder model, to replace the whole model.

After that, I tried to modify my code to make it acceptable for the T5 model. However, I got another problem: a relatively good results on the training part but it would generate out some text which totally unrelated to the input text and resulting in a very low BLEU score. So, I do more research on T5 model and found that the forward function of T5-base model would automatically create the correct decoder_input_ids. Thus, I changed the way I preprocess the data in HW5 and may it more fit with T5-base model. This gave me a better result than the method I used above.

3 Methods

I am using T5 model as my pre-trained model. I fine-rune the model for translate English to German task. And I am using BLEU as my metric for model evaluation.

3.1 Dataset

The dataset I used is wmt18. It has 8 different subsets that can used for translation task. Each subset contains English and one other language. I chose the English and German subset, ‘de-en’ as my data.

The Table 1 shows the size of the dataset in English and German.

Set		Example
Train		42271874
Validation		3004
Test		2998

Table 1: WMT18 ‘de-en’

3.2 Model

Transfer Text-to-Text Transformer, called T5, is a Transformer encoder-decoder model. It can convert all NLP problems into a text-to-text format. I was training the T5 model in supervised fashion. In this setup, I need an input sequence and a target sequence that are a standard sequence-to-sequence input-output mapping.

T5 is using cross entropy to calculate the loss of training. Cross entropy is showing the average number of total bits to represent an event from Q instead of P. The formula (1) shows how cross entropy calculate.

$$Cross\ Entropy = -\sum p(x_i) \cdot \log\big(q(x_i)\big) \quad (1)$$

3.3 Evaluation

I am using Beam Search to do the sampling and using BLEU as metric of my model.

Beam Search is an improvement on the greedy strategy. From example, in Figure 1, we assume a word list of size 5 and contents A, B, C, D, and E. Beam size is 2 means that each timestep will retain two sequences of optimal conditional probability up to the current step. In the first timestep, A and C are the best two, so current two sequences are [A] and [C].

The second step will continue to generate based on these two results. In branch A, 5 candidates can be obtained, [AA], [AB], [AC], [AD], [AE]. In the same way, 5 candidates can be obtained in C. At this point, the 10 will be ranked uniformly, and the best two will be reserved, namely [AB] and [CE] in the figure.

This process is repeated until an end character is reached or the maximum length is reached. The sequences with the highest scores would be finally output.

Also, when the Beam size equal to 1, it would become greedy search.

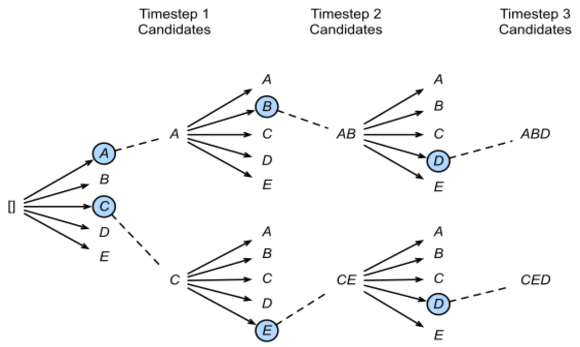

Figure 1: A figure with an example of Beam Search

BLEU stands for bilingual assessment Understudy. Understudy is to evaluate every output of machine translation in place of human. It is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. What Bleu Score does is, given a machine-generated translation, it automatically calculates a score that measures how good the machine translation is.

The formula (2) shows the calculation of BLEU. BLEU's prototype system adopts uniform weighting, that is, Wn=1/N. The upper limit of N is 4, that is, only 4-gram accuracy is counted at most. Among them, BP is the brevity penalty factor, which penalizes the length of a sentence too short to prevent the tendency of training results to short sentences. And Pn, which is based on n-gram accuracy.

$$BLEU = BP * \exp(\sum w_n \log P_n) \quad (2)$$

3.4 Setup

At first, I load the dataset wmt18 and use random to check that the data is what I want to use. I am using T5tokenizer and T5ForConditionalGeneration to load the model. Using T5ForConditionalGeneration to load the model weights would give me a T5 model with a language modeling head on top.

Then, I preprocess the data to make it fit with the model, and I am using AdamW as my optimizer. AdamW is the Adam optimizer with L2 regularization. It limits the value of parameter that would not be too large. The Figure 2 shows the calculation of AdamW.

Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

given $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$
initialize time step $t \leftarrow 0$, parameter vector $\boldsymbol{\theta}_{t=0} \in \mathbb{R}^n$, first moment vector $\boldsymbol{m}_{t=0} \leftarrow \boldsymbol{0}$, second moment vector $\boldsymbol{v}_{t=0} \leftarrow \boldsymbol{0}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$
repeat
  $t \leftarrow t + 1$
  $\nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow \text{SelectBatch}(\boldsymbol{\theta}_{t-1})$ ▷ select batch and return the corresponding gradient
  $\boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1}) + \lambda\boldsymbol{\theta}_{t-1}$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{g}_t$ ▷ here and below all operations are element-wise
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{g}_t^2$
  $\hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t/(1-\beta_1^t)$ ▷ β_1 is taken to the power of t
  $\hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t/(1-\beta_2^t)$ ▷ β_2 is taken to the power of t
  $\eta_t \leftarrow \text{SetScheduleMultiplier}(t)$ ▷ can be fixed, decay, or also be used for warm restarts
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta_t \left(\alpha \hat{\boldsymbol{m}}_t/(\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon) + \lambda\boldsymbol{\theta}_{t-1}\right)$
until *stopping criterion is met*
return optimized parameters $\boldsymbol{\theta}_t$

Figure 2: AdamW

Final, I train my model with the preprocess data and output the results.

4 Analysis and Results

From Table 2, you can find that I am training my t5-base model with 50k steps and the random initialization model with 100k.

And I use batch size 8 for the t5-base model since once I have a larger batch size, the GPU would run out of memory. But for the Random Initialization model, I do not have that kind of problem. So that I was using batch size 16 for it.

Also, I am using 3e-4 as my learning rate for both of my model.

From Figure 3 and Figure 4, we can see the train loss in low and the word accuracy is relatively high for the t5-base model. For the random initialization model, we can see the train loss is still going down and the word accuracy still going up.

Figure 3: Train Loss

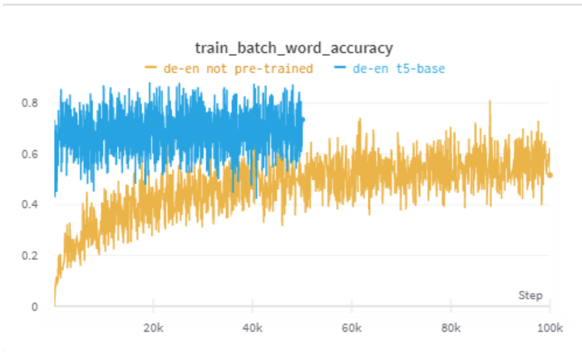

Figure 4: Train Batch Word Accuracy

Also in the Figure 5, the BLEU of t5-base model is relatively high compared with the random initialization model. That means this model still can be improve with more and more steps but not effective as we are using the pre-trained model.

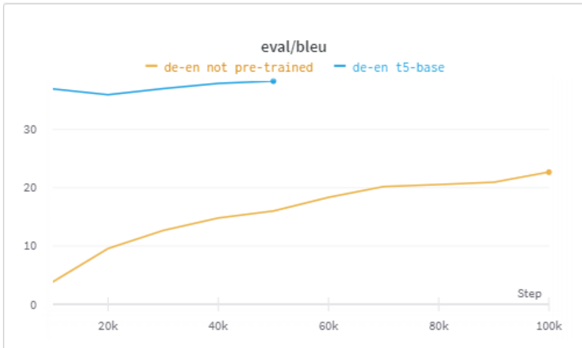

Figure 5: BLEU

I am compared the results my model with the results from other models which using the same dataset wmt18. Based on the Table 2, I thought that I have a good result of my work with t5-base model.

MODEL	BLEU	BATCH	STPES
T5-base	38.3	8	50k
Random Initialization	22.6	16	100k
opus-mt-tc-base-gmw-gmw	36.1		
Multi-pass backtranslated adapted transformer	29.0		

Table 2: results of models

Conclusion

In this project, I learn that how to use a pre-trained component within my own work. I got a BLEU score of 38.3 on the model with T5-base model, a pre-trained encoder-decoder model. And BLEU score of 22.6 on the random initialization model which can be improve more with more training step. And I compared those results with others and get a relatively good answer for my work.

Furthermore, I learned a lot during this project. I believe I have a better understanding of Transformer. And I may do more research and try to do better after.