Text Generation ANN PA3 Report

Add a description...

Eren Zhao

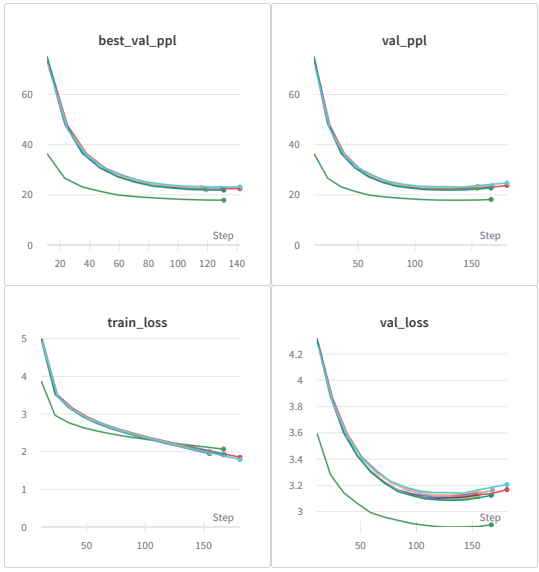
Type '/' for commands

Header 1 Only

基础实验

原始框架中，当 validation set 的 loss 超 Convert to WYSIWYG oss 后，训练会立刻被中断。我修改了这一策略，当连续三次 validation set 的 loss 超过 best validation loss 后，终止训练。这样的提前终止方法一定程度上防止了过拟合。

为了阐释训练结果，先进行图例说明：对于 scratch 模型，我们将其命名为 3_128 与 12_64，代表模型训练时的层数与 batch_size。而 finetune 模型，我们将其命名为 primary_bs128 与 full_bs58，代表 pre_trained checkpoint 为 3 层，batch_size 为 128；pre_trained checkpoint 为 12 层，batch_size 为 58。

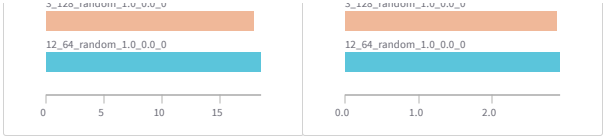


Import panel Add panel

将对应的结果列举在下表：

模型 / 指标	val_ppl	val_loss	Train_loss
3_128	24.095	3.228	2.112
12_64	25.069	3.23	1.843
primary_bs128	19.604	2.829	1.968
full_bs58	15.454	2.829	1.945

此外，将四个模型分别采用 random 策略与 temperature = 1.0 进行测试，测试结果如下：



Import panel Add panel

BLEU 指标，scratch 与 pre-trained

可以见得：

1. 从 perplexity 而言：finetune 模型在 validation set 和 test set 上的 perplexity 都显著低于 scratch 模型，可以见得 pretrain + finetune 的策略在 perplexity 上具有一定优势；
2. 在 training set 上，scratch 与 finetune 的收敛效果相近。我们可以认为，在对训练集的拟合任务上，transformer 本身的能力已经很强劲了（这一点在 TODO 详细讨论）。不过，在 test set 上的差异还是说明了预训练本身能够提高模型的泛化能力；
3. 在基本参数下，finetune 模型与 scratch 模型的 BLEU 指标区别显著低于 ppl 指标，我个人认为这体现了 transformer 本身的强大，然而也反映了 BLEU 指标对于泛化性的反馈是不到位的；

具体而言，BLEU 比较候选译文和参考译文里的 n-gram（实践中从 unigram 取到 4-gram） 的重合程度，重合程度越高就认为译文质量越高。选不同长度的 n-gram 是因为，unigram 的准确率可以用于衡量单词翻译的准确性，更高阶的 n-gram 的准确率可以用来衡量句子的流畅性。

这是一个**只看中准确率**的指标，更关心候选译文里的多少 n-gram 是正确的（即在参考译文里也出现），而不在乎召回率（参考译文里有哪些 n-gram 在候选译文中并未出现）。不过这不算特别严重的问题，因为 BLEU 原论文**建议大家的测试集里给每个句子配备 4 条参考译文**，这样就可以减小语言多样性带来的影响（然而很多机器翻译的测试集都是只有 1 条译文）；另外还有 brevity penalty 来惩罚候选译文过短的情况（候选译文过短在机器翻译中往往意味着漏翻，也就是低召回率）。

但总的来说，学界普遍认为 **BLEU 指标偏向于较短的翻译结果**，因为 brevity penalty 没有想象中那么强。

指标意义

具体到 BLEU 的每一项而言，指标的意义如下：

1. forward BLEU-4 主要反映语句的流利程度，也即单个句子的准确程度；
2. backward BLEU-4 主要反映生成的所有句子的丰富度，也即模型能够生成更多多变的结果；
3. harmonic BLEU-4，也即 chart 中的 forward backward BLEU-4，同时取决于 forward BLEU-4 和 backward BLEU-4；是准确度和丰富度的一种权衡。当然，仅仅通过 fw-bw-bleu-4 来参考句子的生成质量是不合理的，还需要通过本次报告内的 case study 来人工 evaluate。

指标缺点

总而言之，BLEU 具有如下缺点：

1. 不考虑语言表达（语法）上的准确性；
2. 测评精度会受常用词的干扰；
3. 短译句的测评精度有时会较高；

Decoding Strategies

对比 inference 方法

对于每个模型，我按照如下的配置测试了 inference 的超参数。

```
experiments = [
    ("random", 1, 0, 0),
    ("random", 0.85, 0, 0),
    ("random", 0.7, 0, 0),
    ("top-p", 1, 0.9, 0),
    ("top-p", 1, 0.8, 0),
    ("top-p", 1, 0.7, 0),
    ("top-p", 0.7, 0.9, 0),
    ("top-k", 1, 0, 40),
    ("top-k", 1, 0, 30),
    ("top-k", 1, 0, 20),
    ("top-k", 0.7, 0, 40),
]
# inference 策略, temperature, top-p, top-k
```

我将实验设定的 3 层 ckpt 命名为 primary.tar，12 层 ckpt 命名为 full.tar。为了完成扩展要求 2，我还设定了如下的 extract 方案：

```
extraction_dict = {1: "first", 2: "last", 3: "skip"}
# 分别代表每种 extract 方案的模型名称后缀
mappings = [
    {"0": "0", "1": "1", "2": "2"},
    {"0": "9", "1": "10", "2": "11"},
    {"0": "0", "1": "5", "2": "11"},
```



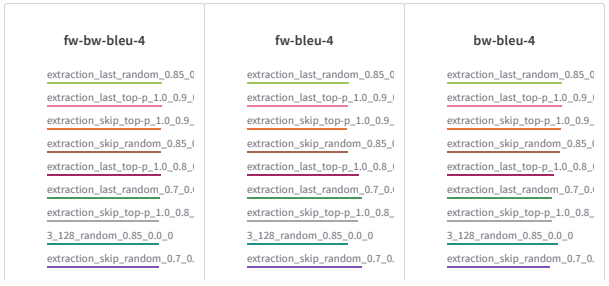
```
]
# 代表每种方案从 full.tar 中抽取出的对应层数
```

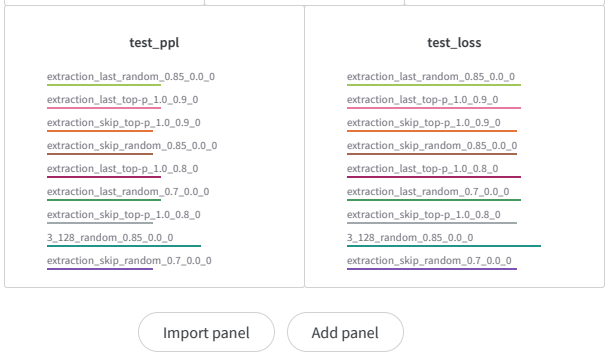
最终，我设置了如下的 7 个模型：

```
--num_layers=3 --batch_size=128 # 此即 3_128 模型
--num_layers=12 --batch_size=64 # 此即 12_64 模型
--pretrain_dir=./ckpt/full.tar --batch_size=58 # 此即 full_58 模型
--pretrain_dir=./ckpt/primary.tar --batch_size=128 # 此即 primary_128 模型
--pretrain_dir=./ckpt/full.tar --extract_layer=1 --batch_size=128 # 此即 extract_first 模型
--pretrain_dir=./ckpt/full.tar --extract_layer=2 --batch_size=128 # 此即 extract_last 模型
--pretrain_dir=./ckpt/full.tar --extract_layer=3 --batch_size=128 # 此即 extract_skip 模型
```

在此基础上，我对于 7 个模型都进行了 9 个 inference 实验，并且实验命名原则为 {model_name}_{strategy}_{temperature}_{top-p}_{top-k}。注意，为了报告的直观，本部分的 chart 中每一个测试指标只展示了 fw-bw-bleu-4 最高的 10 个实验，具体的结果与其他实验的结果可以双击 chart 下方的 run set 内具体查看。

总体得分较高的结果如下：



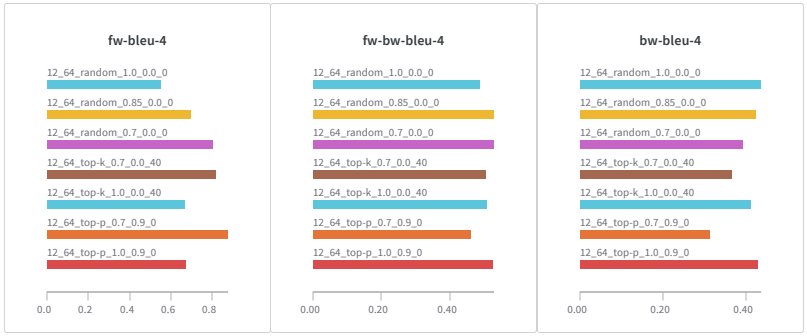


Import panel　Add panel

我们进一步详细讨论：

1. 对于 temperature 参数：在三种策略下，temperature 参数减小都会导致 fw-bleu-4 增大，bw-bleu-4 减小，反映出 temprature 减小会加强单个句子的质量，但是会降低句子的多样性。实际上，temperature 参数作用与解码器的输出层。输出层后通常会通过 softmax 函数来将输出概率归一化，通过改变 temperature 可以控制概率的形貌。当 temperature 较大的时候，概率分布趋向平均，随机性增大；当 temperature 较小的时候，概率密度趋向于分散，高概率者出现的可能更强，随机性降低。

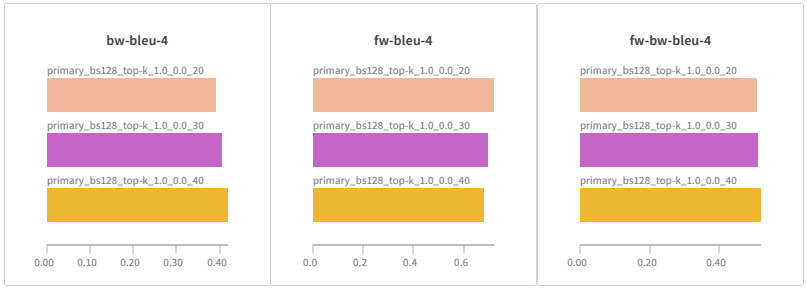
$$p_i = \frac{\exp(y_i/T)}{\sum \exp(y_i/T)}$$



Import panel Add panel

2. 对于 p 参数，在 top-p decoding 时，p 减小都会导致 fw-bleu-4 增大，bw-bleu-4 减小，反映出 p 减小会加强单个句子的质量，但是会降低句子的多样性。具体而言，Top-p (nucleus) sampling 是 Ari Holtzman et al. (2019) 提出的算法。从使得累计概率超过 p 的最小候选集里选择单词，然后算这些单词的概率分布。这样候选单词集的大小会随下一个单词的概率分布动态增加和减少。当 p 减小时，模型筛选出更少的单词集合，提升了单一句子质量，然而选项更少，句子的丰富度降低；反之，p 增大时，模型筛选出更多的单词集合，可能会降低单一句子质量，然而选项更

token。但 Top-k Sampling 存在的问题是，常数 k 是提前给定的值，对于长短大小不一，语境不同的句子，我们可能有时需要比 k 更多的 tokens。



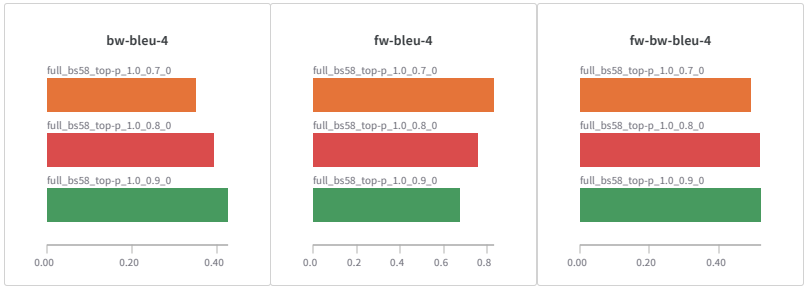
Import panel Add panel

Perplexity

我们首先回顾 perplexity 的数学意义：

Perplexity，即复杂度，是语言模型的效果好坏的常用评价指标。在一个测试集上得到的perplexity越低，至少说明模型的拟合能力效果越好，计算 perplexity 的公式如下：

多，能产生更加丰富的句子，当 p = 1.0 时，实际上已经等同于 random 策略。另外，在实际研究中，往往会为 top-p 策略加入 minimal candidate 参数，避免候选的单词集大小过小。



Import panel Add panel

3. 对于 k 参数，在 top-k decoding 时，k 减小都会导致 fw-bleu-4 增大，bw-bleu-4 减小，反映出 k 减小会加强单个句子的质量，但是会降低句子的多样性。top-k 方法比上一个更加简单。在 decoding过程中，从 $P\left(x \mid x_{1:t-1}\right)$ 中选取 probability 最高的前 k 个 tokens，把它们的 probability 加和得到 $p' = \sum P\left(x \mid x_{1:t-1}\right)$，然后将 $P\left(x \mid x_{1:t-1}\right)$ 调整为 $P'\left(x \mid x_{1:t-1}\right) = P\left(x \mid x_{1:t-1}\right)/p'$，其中 $x \in V^{(k)}$，最后从 $P'\left(x \mid x_{1:t-1}\right)$ 中 sample 一个 token 作为 output

$$\begin{aligned}\text{perplexity}(S) &= p\left(w_1, w_2, w_3, \ldots, w_m\right)^{-\frac{1}{m}} \\ &= \sqrt[n]{\prod_{i=1}^{m} \frac{1}{p\left(w_i \mid w_1, w_2, \ldots, w_{i-1}\right)}}\end{aligned}$$

简单来说，perplexity 刻画的是语言模型预测一个语言样本的能力，比如已经知道了 $\left(w_1, w_2, w_3, \ldots, w_m\right)$ 这句话会出现在语料库之中，那么通过语言模型计算得到这句话的概率越高，说明语言模型对这个语料库拟合的越好。

本次任务和大多语言模型的训练相同，采用了 perplexity 的对数表达形式：

$$\log\left(\text{perplexity}\left(S\right)\right) = -\frac{1}{m}\sum_{i=1}^{m} p\left(w_i \mid w_1, w_2, \ldots, w_{i-1}\right)$$

相比较乘积求平方根的方式，采用加法的形式可以加速计算，同时避免概率乘积数值过小而导致浮点数向下溢出的问题。

在数学上，log perplexity 可以看作真实分布与预测分布之间的交叉熵 Cross Entropy，交叉熵描述了两个概率分布之间的一种距离，假设 x 是一个离散变量，$u(x), v(x)$ 是两个与 x 相关的概率分布，那么 u, v 之间的交叉熵的定义是分布 u 下 $-\log(v(x))$ 的期望值：

$$H(u, v) = E_u[-\log v(x)] = -\sum_x u(x) \log v(x)$$

我们把 x 看作是单词，$u(x)$ 表示每个位置上单词的真实分布。

$$u\left(x \mid w_1, w_2, \ldots, w_{i-1}\right) = \begin{cases} 1, x = w_i \\ 0, x \neq w_i \end{cases}$$

$v(x)$ 是模型的预测分布 $p\left(w_i \mid w_1, w_2, \ldots, w_{i-1}\right)$，那么即有：

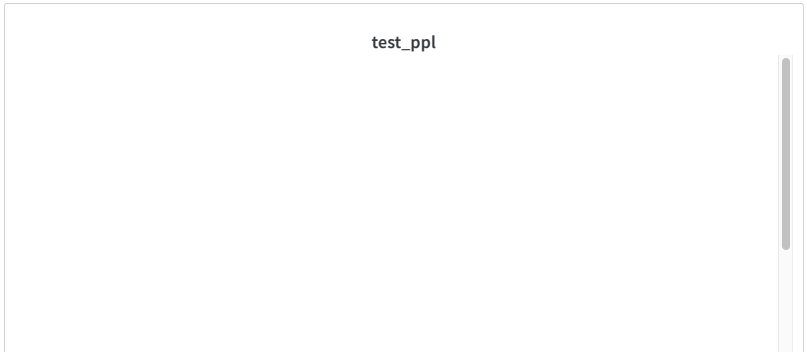
$$
\begin{aligned}
H(u, v) &= -\sum_x u(x) \log v(x) \\
&= -\frac{1}{m} \sum_{i=1}^{m} \left(\sum_x u(x \mid w_1, \ldots, w_{i-1}) \, p(w_i \mid w_1, w_2, \ldots, w_{i-1}) \right) \\
&= -\frac{1}{m} \sum_{i=1}^{m} \left(1 \times p(w_i \mid w_1, w_2, \ldots, w_{i-1}) + \sum_{x \neq w_i} 0 \times p(w_i \mid w_1, w_2, \ldots, w_{i-1}) \right) \\
&= -\frac{1}{m} \sum_{i=1}^{m} p(w_i \mid w_1, w_2, \ldots, w_{i-1}) \\
&= \log \left(\text{perplexity}^{2}(S) \right)
\end{aligned}
$$

因此 log perplexity 和交叉熵实质上是等价的。

实际上，我们可以发现，perplexity 仅仅是计算某个句子存在时它被模型生成的条件概率，故而 perplexity 仅仅与 temperature 有关，和 decoding strategy 是无关的。故而我固定 strategy 为 random decoding，采用了如下的实验序列：

```
temperatures = [0.4, 0.55, 0.7, 0.85, 1.0, 1.15, 1.3, 1.45, 1.6]
models = ["./train_ckpt/3_128.tar", "./train_ckpt/12_64.tar"]
```

./train_ckpt/3_128.tar 得到的实验结果如下：



综上可见，perplexity 对于 temperature 参数呈现典型的 U 型曲线，过大过小的 temperature 对于 perplexity 都有负面影响，过小的 perplexity 尤其如此；两模型最优的 perplexity 均在temperature = 1.15 时取得。temperature 对于 perplexity 的影响较为复杂。模型完成预测，生成 logits 后，如果某个单词的 logit 最大，此时减小 temperature 有利于降低 perplexity；而某个单词的 logit 不是最大时，反而减小 temperature 有利于降低 perplexity。总之，temperature 对于 perplexity 的影响有待具体分析。

Case Study

为了方便模型的横向对比，我选择了 12_64 模型（Tfmr-scratch）与 full_bs58 模型（Tfmr-finetune）进行对比，同时三种 decoding 方式按照要求选用 random_1.0_0.0_0、random_0.85_0.0_0、random_0.7_0.0_0、top-p_1.0_0.9_0、top-p_0.7_0.9_0、top-p_1.0_0.7_0、top-k_0.7_0.0_40、top-k_1.0_0.0_40、top-k_1.0_0.0_20。

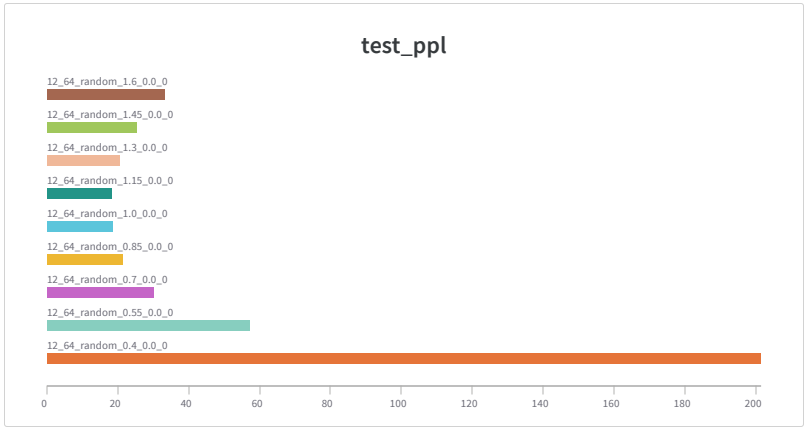
Scratch

首先展示 12_64 模型效果：

```
#-------------  12_64_random_1.0_0.0_0  -------------
A black bird standing next to a bunch of jacket trough .
Two very young buses standing beside coast bumper many tall bars and farm Wall .
 proposition art maximal rather judge graffiti flying trails inStates in Cutler plants .
Man traveling mores myself down aele fly at night .
Image of writing on overflowingariPolitics for guests to agreeing outside .
Woman sitting on wooden bench reading by chairs in the goal and walk .
```

Import panel　Add panel

./train_ckpt/12_64.tar 得到结果如下：



Import panel　Add panel

```
A cat crossing a sidewalk near a house near a street .
An abandoned bayite sitting on a green field where bathroom .
We are see green over a black and hat set fashioned on there stove , and a Clare .
The coffee shop is pan with several messy guard recently wood bench together . One racing .


#-------------  12_64_random_0.85_0.0_0  -------------
A close up of two red bus stop with cars in the background .
A green transit bus with a bike on the street .
A woman wearing a leather hat sitting on a bench .
A giraffe walking past a fence and a wooden fence .
A couple of white busses parked along the side of the road beside a tree .
A giraffe looking over a fence full of other animals .
A man sits on a park bench with a dog laying in the background .
A woman sitting on a chair in a large bunch of suit .
A bright yellow fire hydrant in front of a large building .
A bus that is sitting in front of a bus stop .


#-------------  12_64_random_0.7_0.0_0  -------------
A close up of two red bus stop with cars in the background .
A green transit bus is parked on the side of the road .
A woman wearing a leather hat sitting on a bench .
A giraffe walking past a tree branch on a sunny day .
A couple of giraffe walking through a dirt field .
A giraffe looking over a fence at something of zoo .
A man sits on a park bench with a dog is reading .
A woman sitting on a wooden bench next to a dog .
A couple of giraffe walking across a street in a park .
A bus that is sitting in front of a bus stop .
```

```
#-------------  12_64_top-p_1.0_0.9_0  -------------
A close up of two red bus stop with cars in the background .
Two buses are parked next to each other in front of a tree .
A woman wearing a leather hat sitting on a bench .
A giraffe walking past a fence and lots of palm trees .
A couple of white busses parked along the side of the road .
A giraffe looking over a fence eating leaves off an enclosed area .
A man feeding a giraffe while another boy on a tree .
A woman sitting on a wooden bench next to a traffic light .
A bright yellow fire hydrant in front of a large building .
A bus that is sitting in front of a bus stop .


#-------------  12_64_top-p_0.7_0.9_0  -------------
A close up of two red bus stop with cars in the background .
A green bench sitting next to a tree in a park .
A woman sitting on a bench near a tree outside .
A giraffe walking past a tree branch on a sunny day .
A couple of giraffe walking through a dirt field .
A giraffe is standing with two other zebras and a baby giraffe behind them .
A man sits on a bench while reading a magazine .
A woman sitting on a bench in a park with a dog sits on her legs phone .
A couple of giraffe walking across a dry grassy field .
A bus that is sitting in front of a bus stop .


#-------------  12_64_top-p_1.0_0.7_0  -------------
A close up of two red bus stop with cars in the background .
A city street with lots of traffic and a pedestrian -a bus with tall buildings .
A woman wearing a leather hat sitting on a bench .
A giraffe walking past a fence and lots of palm trees .
A couple of white busses parked in front of trees .
A giraffe is eating some leaves off to the neck .
A man sits on a bench while reading a magazine .
A woman sitting on a bench in a large park .
A very cute looking giraffe over its trees in the wild .
A bus that is sitting in front of a bus stop .


#-------------  12_64_top-k_0.7_0.0_40  -------------
A close up of two red bus stop with cars in the background .
A green transit bus is parked on the side of the road .
A woman wearing a leather hat sitting on a bench .
A giraffe walking past a tree branch on a sunny day .
A couple of giraffe walking through a dirt field .
A giraffe looking over a fence at something of zoo .
A man sits on a park bench with a dog is reading .
A woman sitting on a wooden bench next to a dog .
A couple of giraffe walking across a street in a park .
A bus that is sitting in front of a bus stop .


#-------------  12_64_top-k_1.0_0.0_40  -------------
A close up of two red bus stop with cars in the background .
Two buses are parked next to each other in front of a tree .
A woman wearing a leather hat sitting on a bench .
A giraffe walking past a fence and lots of palm greenery .
A couple of white busses parked along the side of the road beside a mountain .
A giraffe looking over a fence eating leaves off the camera .
A man feeding something on a bench with a sign .
A woman sitting on a chair in a large pen .
A very cute looking giraffe over its trees in snow .
A bus that is sitting in front of a bus stop .


#-------------  12_64_top-k_1.0_0.0_20  -------------
A group of lambs standing around a white and blue airplane .
Two buses are parked next to each other in front of a tree .
A woman wearing a leather hat sitting on a bench .
A giraffe walking past a fence and a wooden fence .
A couple of white busses parked along the side of the road beside a mountain .
A giraffe looking over a fence eating leaves off the camera .
A man feeding something on a bench with a sign .
A woman sitting on a chair in a large pen .
A very cute looking giraffe over its trees in snow .
A bus that is sitting in front of a bus stop .
```

Pretrained

其次展示 full_bs58 模型效果：

```
#-------------  full_bs58_random_1.0_0.0_0  -------------
A close up of a red bus driver with copter on a snowy night .
Two buses are traveling across a dirt trail .
A woman wearing a leather jacket sitting on a bench with her family in the background .
A giraffe walking past a fence and a fence back in its habitat in an outdoors setting .
A couple of white and blue buses sitting along a street .
A bus driving down a corner near red buses .
A man sits with an airplane on the runway .
An airplane parked on the tarmac at an airport .
A bright yellow fire hydrant in front of a door .
A bus that is sitting next to another bus on the side of the road .


#-------------  full_bs58_random_0.85_0.0_0  -------------
A close up of a red bus and a cop car .
A green transit bus parked along a sidewalk in front of a tree .
A woman wearing a leather jacket sitting on a bench .
A giraffe walking past a fence and a fence back in its habitat .
A couple of giraffe walking through grassy area next to tree .
A bus driving down a street near cars and buildings .
A man sits on a park bench with a motorcycle .
A woman sitting on top of a bench by herself .
A bright yellow fire hydrant in front of a large building .
A bus that is sitting in front of a truck next to a bus stop .


#-------------  full_bs58_random_0.7_0.0_0  -------------
A close up of a red bus and a cop car .
A green transit bus parked on the side of the road .
A woman sitting on a bench near a cup of coffee .
A giraffe walking past a fence and a fence post .
A couple of giraffe walking through a forest , a tree , and a few other animals .
A bus driving down a street near cars and buildings .
A man sits on a bench while waiting for the bus .
A woman sitting on top of a bench by herself .
A man is sitting on a bench with his dog .
A bus that is sitting in front of a truck .


#-------------  full_bs58_top-p_1.0_0.9_0  -------------
```

```
A close up of a red bus driver with a woman on a leash .
Two buses are traveling across a dirt trail .
A woman wearing a leather jacket sitting on a bench with her family in the background .
A giraffe walking past a fence and a fence on a sunny day .
A couple of white and blue buses sitting in front of a bunch of buildings .
A bus driving down a street near cars and buildings .
A man sits on a park bench with a motorcycle .
An airplane parked on the tarmac at an airport .
A bright yellow fire hydrant in front of a door .
A bus that is sitting next to another bus on the side of the road .


#-------------  full_bs58_top-p_0.7_0.9_0  -------------
A close up of a red bus and a red fire hydrant .
A green bench sitting on top of a grassy hill .
A woman sitting on a bench near a green bench .
A giraffe walking through a field next to a tree .
A couple of giraffe walking through a forest , a tree , and a few other animals .
A bus driving down a street near a traffic light .
A man sits on a bench while reading a book .
A woman sitting on top of a bench by a traffic light .
A man is sitting on a bench with his dog .
A bus that is sitting in front of a truck .


#-------------  full_bs58_top-p_1.0_0.7_0  -------------
A close up of a red double decker bus .
A green bench sitting on top of a grassy hill .
A woman sitting on a bench near a green bench .
A giraffe walking through a wooded area near a stone wall .
A couple of white and blue buses sitting in front of a store .
```

```
A bus driving down a street near tall buildings .
A man sits on a bench while reading a book .
A woman sitting on a bench in a park .
A bench sitting in front of a forest and some shrubs .
A bus that is sitting in front of a truck .


#-------------  full_bs58_top-k_0.7_0.0_40  -------------
A close up of a red bus and a red fire hydrant .
A green transit bus parked on the side of the road .
A woman sitting on a bench near a green bench .
A giraffe walking past a fence and a fence post .
A couple of giraffe walking through a forest , a tree , and a few other animals .
A bus driving down a street near cars and buildings .
A man sits on a bench while waiting for the bus .
A woman sitting on top of a bench by herself .
A man is sitting on a bench with his dog .
A bus that is sitting in front of a truck .


#-------------  full_bs58_top-k_1.0_0.0_40  -------------
A close up of a red bus driver with a woman on a leash wearing sunglasses .
Two buses are traveling across a dirt trail .
A woman wearing a leather jacket sitting on a bench with her family in the background .
A giraffe walking past a fence and a fence on a sunny day in a sunny part of the country .
A couple of white and blue buses sitting along a street .
A bus driving down a street near red buses .
A man sits with an airplane on the runway .
An airplane parked on the tarmac at an airport .
A bench near a grassy plain with trees in the background .
A bus that is sitting next to another bus on the side of the road .
```

```


#-------------  full_bs58_top-k_1.0_0.0_20  -------------
A group of men standing on the street with their bags on a bench .
Two buses are traveling across a dirt trail .
A woman wearing a hat sits on a bench outside .
A giraffe walking past a fence and a fence on a sunny day in a sunny part of the country .
A couple of white and blue buses sitting along a street .
A bus driving down a street near cars and buildings .
A man sits with an airplane on the runway .
An airplane parked on the tarmac at an airport .
A bench near a wall with some buildings in the background .
A bus that is sitting next to another bus on the side of the road .
```

分析

1. 两组生成结果在语法上都有不少的错误，然而整体上 top-p 解码方式比 random 解码方式生成的单句质量更高，读起来更为流畅。比如 12_64_top-p_0.7_0.9_0 中 A giraffe walking past a tree branch on a sunny day 这样的语句几乎没有错误，而且生成的场景也符合常理；而 12_64_random_1.0_0.0_0 生成的语句 An abandoned bayite sitting on a green field where bathroom 本身语句不完整，且句意也不符合常理。自然，top-p decoding 会比 random decoding 缺少创造性，不过这在 10 个 case study 中难以体现。
2. 同理，在 top-k decoding 中，k 从 40 减少到 20 后，生成的单句质量也有了一定的提升。譬如 12_64_top-k_1.0_0.0_20 生成有 Two buses are parked next to each other in front of a tree，而 12_64_top-k_0.7_0.0_40 生成有 A man sits on a park bench with a dog is reading。显然，前者的语句质量更高。
3. 较小的 temperature 参数也能够增大单句质量，譬如 12_64_random_0.7_0.0_0 生成有 A giraffe walking past a tree branch on a sunny day，而 12_64_random_0.85_0.0_0 生成了 A giraffe walking past a fence and a wooden fence。前者更有语句意义，而后者更像是无意义的重复。
4. 总体上，finetune 模型的生成结果优于 scratch 模型。自然，模型进行 trans-learning 后理应具有更好的泛化性和生成能力。当然，预训练也有自身的缺点，譬如高昂的训练代价。
5. 综上所述，case study 的结果与前文的分析吻合。并且，full_bs58_top-p_0.7_0.9_0 生成的语句我个人认为效果最佳，这与其前向 BLEU 分数最高相符。

最终提交

1. 在不考虑扩展要求的前提下，我选择的提交模型是 3_128_random_0.85_0.0_0，相应指标如下：

```
fw-bw-bleu-4 = 0.5513
fw-bleu-4 = 0.7347
bw-bleu-4 = 0.4412
test_perplexity = 20.687
test_loss = 3.03
best_epoch = 10
batch_size = 128
best_val_ppl = 24.094548493213612
train_loss = 1.9681225368532085
training_epoch = 13
val_loss = 3.230439085793495,
val_ppl = 25.290759346472576
```

相应的生成结果为 `./output/3_128_random_0.85_0.0_0.txt`。

2. 考虑扩展要求后，我选择提交的模型是 extraction_skip_top-p_1.0_0.8_0，模型相应指标如下：

```
fw-bleu-4 = 0.8124162087120956
fw-bw-bleu-4 = 0.5538688840087167
bw-bleu-4 = 0.4201562843460485
```

```
test_ppl = 14.183500149310456
test_loss = 2.6520793276906014
best_epoch = 11
batch_size = 110
best_val_ppl = 16.81304284434261
train_loss = 2.0743113493515275
training_epoch = 14
val_loss = 2.8365362881183622
val_ppl = 17.056584009739563
```

可以见得，考虑了 extract pretrain model 后，模型的泛化能力有了显著提升，相应的生成结果为 `./output/extraction_skip_top-p_1.0_0.8_0.txt`。

扩展问题

Discussion

1. 为什么 multi-head attention 比 single-head attention 的效果好?

   multi-head attention 结构设计能让每个注意力机制通过 Q、K、V 映射到不同的空间去学习特征，去优化每个词汇的不同特征部分，从而均衡同一种注意力机制可能产生的偏差。每个 attention head 最终只关注最终输出序列中一个子空间，彼此间独立，让词义拥有更多元的表达来源。实验表明 multi-head 能够显著提升模型效果。

2. **BPE** 分词器与空格分词器的区别?

   空格分词器也即 Word-based Tokenizer，在该分词器作用下，每个词都被分配了一个 ID，从 0 开始，一直到词汇表的大小。该模型使用这些 ID 来识别每个词。

   如果研究者希望利用 Word-based Tokenizer 完全覆盖一种语言，就需要为该语言中的每个词都有一个标识符，这将产生大量的标记。例如，英语中有超过 50 万个单词，所以要建立一个从每个单词到输入 ID 的映射，需要跟踪非常多 ID。此外，像 dog 这样的词与 dogs 这样的词的表示方法不同，模型最初将没有办法知道 dog 和 dogs 是相似的：它将识别这两个词是不相关的。同样的情况也适用于其他类似的词，比如 run 和 running，模型最初也不会认为它们是相似的。

   最后，模型需要一个自定义标记来表示不在词汇表中的单词。这就是所谓的未知标记，通常表示为 [UNK]。如果分词器产生了很多这样的标记，这通常是一个不好的迹象，因为它无法检索到一个词的合理表示，而模型正在沿途丢失信息。制作词汇的一大目标是使标记器尽可能少地将单词标记到未知标记中，而显然，Word-based Tokenizer 将面对大量的未知标记。

   BPE tokenizer 属于 Subword-based tokenizer。BPE 本身是一种简单的数据压缩算法，其中最常见的是将一对连续字节替换为该数据中不出现的字节。BPE 分词器能够达到将常用的单词在一定程度内保留，而生僻单词分解为有意义子段的目的。不常出现的词被分解成不同的 Token 参与训练，避免了大量的未知标记，也降低了词表的复杂度，有利于训练。此外，BPE tokenizer 还可以一定程度上实现 embedding sharing，前文的 dog，dogs，dogge 都可以 share dog 这个字词的 embedding，在神经网络层共享特征，也降低了训练复杂度。

3. Transformer 与 RNN 的对比?

   从模型表现力的角度来看，Transformer 利用三维张量映射来建模一个 language translation 过程是非常优秀的，甚至可以说是现有算力下的极致模型。Transformer 避免了逐单词处理序列的问题，让他本身就具有了双向性。对于 CNN 与 RNN，两个不同的 token 相互作用时，模型的顺序性决定了这两个不同 token 的先后并不影响结果。而对 transformer 而言，两个不同的 token 在彼此视角下的 attention 是不同的。可以如此类比，transformer 当中的两个 token 如同两个人 Alice 和 Bob，Alice 眼中，自己和 Bob 的相互关系大概率和 Bob 眼中自己和 Alice 的相互关系是不同的，这承载了丰富的语义信息，而 CNN 和 RNN 则走向了“我见青山多妩媚，料青山见我当如是”的局限。此外，Transformer 当中的 token attention 相互关系相较 RNN 中沿着 recurrence 链条的相互关系是更加稳定的，应为 RNN 沿着序列遍历时更容易出现梯度消失或者爆炸，而 attention 能够对任意距离的 token 产生更为稳定的结果。当然，这一分析也存在许多漏洞。因为

   此处只涉及了 self-attention 层，而 layernorm 和残差对于 transformer 也非常重要。此外，有些强时序性的任务 RNN 效果要好于 Transformer（因为 transformer 的位置编码太弱），有些局部特征重要的图像任务 CNN 也比 Transformer 更 work。同时，也有一些理论和实验表明，CNN 可以看作是高频滤波器，self-attention 是低频滤波器，所以残差对于 transformer 非常重要。这两种模型其实在理论上的好坏还有很多要验证的地方。虽然 transformer 非常强大，但是仍然需要深入讨论其缺陷性，避免对 transformer 的神化，毕竟 transformer 并非深度学习历史的终结。

   从模型参数数量的角度来看，Transformer Block 可以轻松堆叠起来形成参数量庞大的健壮模型，并可以对每一层施加 auxiliary loss；而 RNN 可能会受到梯度消失等问题的影响从而有一定的限制，不能够将层数堆叠太大。

   从时间复杂度的角度来看，考虑 n 为序列长度，d 为 hidden layer 维度，则 transformer 的复杂度为 $\mathrm{O}\left(\mathrm{n}^2\mathrm{d}+\mathrm{nd}^2\right)$，而RNN的时间复杂度是 $\mathrm{O}\left(\mathrm{nd}^2\right)$，看上去当 n 远超过 d 时，transformer 的复杂度会有显著劣势，然而 transformer 的每一层都能够高效并行，甚至还有 Megatron 这样的工作能够从系统层面优化 transformer layer 的并行计算，而 RNN 囿于模型必然的顺序性，只能够串行计算。总之，现有的加速框架能够弥合 transformer 复杂度上的不足，而相比时间复杂度的缺憾，transformer 的模型强度足够吸引人。

4. 时间复杂度分析

   `use_cache` 的作用实质上和 PA1 中对 MLP 的 cache 意义是一致的，也即避免重复计算。做 inference 时，模型逐 token 进行 decode。使用 cache 后，新增的 token 时可以仅仅对新增的 token 进行计算，之前计算的 cache 结果直接包含着已经计算的 token 信息，可以直接利用。如此避免了对于每个 token 都进行重复的 $\mathrm{O}(\mathrm{n}^2)$ attention 计算。此外，对于 transformer 而言，大概率复杂度的瓶颈在 FFN 层。利用 cache 也能避免 token 反复通过 FFN 层。不过，即便是对单一的 token 做 attention，利用 cache 能够把单个 attention 的复杂度降为 $\mathrm{O}(\mathrm{n})$，整体的复杂度仍旧是 $\mathrm{O}(\mathrm{n}^2)$ 的。

   Multi-header 机制并不影响复杂度，这其实从最后一个扩展实验也可见得。故而，decode $\mathrm{l_t}$ 时，每一次 transformer block 的 attention time complexity 时 $\mathrm{O(td)}$，FF 层的复杂度为 $\mathrm{O}(\mathrm{d}^2)$，故而通过所有的 attention block 的 time complexity 为 $\mathrm{O}\left(\mathrm{B}\left(\mathrm{td}+\mathrm{d}^2\right)\right)$。最后，LM head 层的复杂度为 $\mathrm{O(dV)}$。综上所述，decode $\mathrm{l_t}$ 的 time complexity 是 $\mathrm{O}(\mathrm{d}(\mathrm{Bt}+\mathrm{Bd}+\mathrm{V}))$，对其累计求和，既有 inference 整句的 time complexity 为 $\mathrm{O}(\mathrm{Ld}(\mathrm{Bd}+\mathrm{BL}+\mathrm{V}))$。

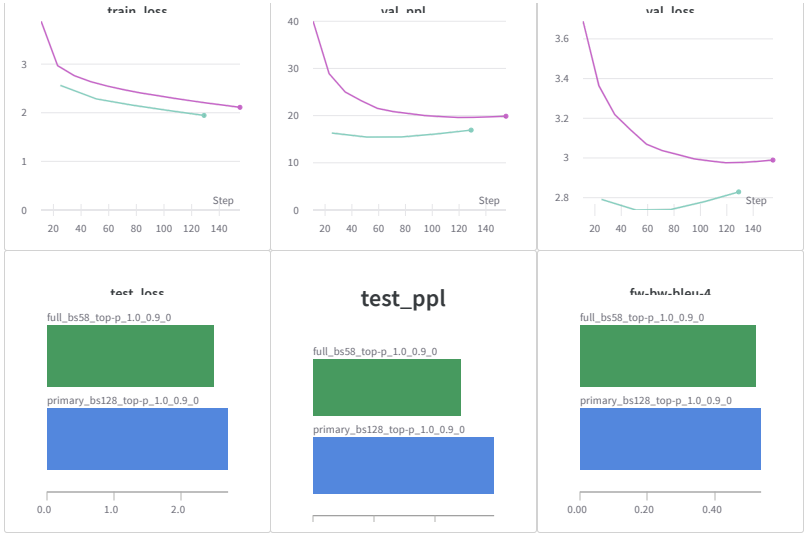
   最后，单层 self-attention 的 time complexity 为 $\mathrm{O}\left(\mathrm{L}^2\mathrm{d}\right)$，而 FFN 的 time complexity 为 $\mathrm{O}\left(\mathrm{Ld}^2\right)$。仅就此，我们可以认为 $\mathrm{L} \gg \mathrm{d}$ 时，attention dominate，而 $\mathrm{d} \gg \mathrm{L}$ 时，FFN dominate。不过，就我的认知所见，在工业界的应用基本是 FFN dominate。

5. 预训练模型的影响?

   从实验结果上看，预训练模型在 perplexity 和 BLEU 上的指标都是更优秀的，而且收敛速度训练稳定度也有明显优势。可以见得，预训练对于提高模型的最终效果是很好的。我认为这和迁移学习的机制有直接联系——迁移学习使得模型的最终效果和收敛速度都有了一定提升。毕竟模型在先前的训练中已经掌握了一定的语言能力，在之后的下游任务实际上做的是 domain transformation，有了一定基础，transform 的效率比 study from scratch 相对高些。此外，本次作业的下游任务数据集并不大，from scratch 的学习难度进一步加深。 当然，预训练的坏处也是明确的，毕竟高昂的预训练成本不容忽略，而且模型究竟是有学习到语言能力从而根本上提高了模型的泛化能力，还是仅仅因为测试集实际上是预训练集的子集导致了模型仅仅依靠拟合就产生了可观的效果，这是值得考究的。毕竟 GPT-3 的工作专门对于 dirty pretrain data 进行了一定讨论，此后在这个领域还衍生出了以 Chain of Thought 为 prompt 的研究热潮。

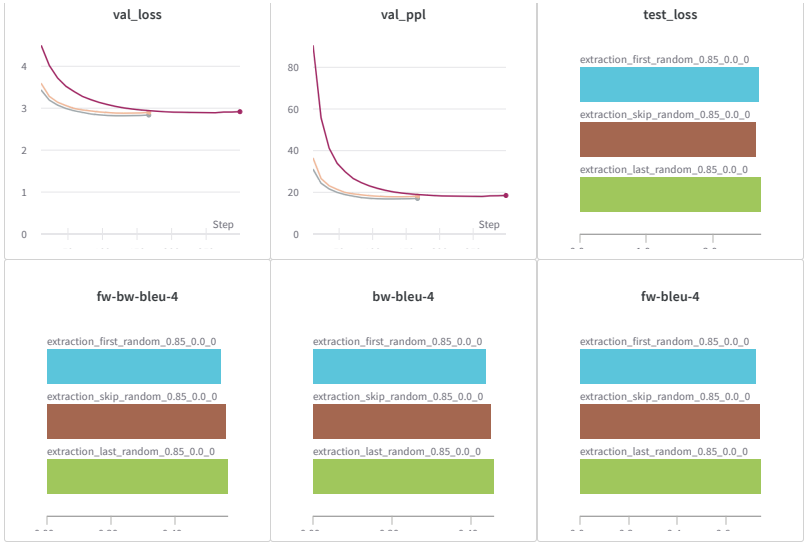
Layer Discussion

为了比对 3 层与 12 层 transformer 的表现能力，我选取了 primary_bs128 模型和 full_bs58 模型进行对比，并且选择了 top-p_1.0_0.9_0 decoding strategy，其相应结果如下：



Import panel Add panel

从实验结果上可以见得，primary_bs128 模型的表现在除了 fw-bw-bleu-4 的五项指标上优于 full_bs58 模型，而 fw-bw-bleu-4 仅仅有微弱的劣势。这证明了 12 层模型相较 3 层模型的确具有更强



Import panel Add panel

大的潜能。

我个人认为，客观上，更大的层数昭示着更强大的潜力，在面对复杂任务时，结合 于residual connection、layer normalization、AdamW 等方法，能够有更强更稳定的学习能力；不过在没有这些方法辅助的情况下，并不一定更高的层数表现必然更好，在第一次实验中双层 MLP 不如单层 MLP 即是典例。

其次，仅就本次实验而言，在 GPT-2 的预训练过程中，3 层模型最后一层所接入的并非 multi head，不过在 finetune 阶段，最后一层被人为接入 multi head；第三层所理解的语义信息是较低的，然而需要 handle 一些高层的信息，这是自然影响了模型的表现力；而 12 层模型越靠后的层所理解的信息越高层，处理高层语义信息自然更加合适。

Extraciton Study

为了比对出 full.tar ckpt 不同层习得的语义信息，我按照如下方案进行了 extraction study：

```
extraction_dict = {1: "first", 2: "last", 3: "skip"}
# 分别代表每种 extract 方案的模型名称后缀
mappings = [
    {"0": "0", "1": "1", "2": "2"},
    {"0": "9", "1": "10", "2": "11"},
    {"0": "0", "1": "5", "2": "11"},
]
# 代表每种方案从 full.tar 中抽取出的对应层数
```

同时我选择用 random_0.85_0.0_0 策略来对模型进行测试。

相应的结果如下图所示：

综合考虑 val_perplexity，BLEU-4 与 test_loss 的指标，我认为用 skip 策略，也即 full ckpt 的 1，6，12 层来初始化 3-layer GPT-2 取得了最好的效果。这非常符合预期，正如前文所述，越低层的 layer 得到越底层的语义信息，越高层的 layer 掌握越高层的语义信息。1，6，12 三层在原本的 ckpt 中将分别会负责提取底层，中层，高层的语义特征。

将三者 extract 出来，initialize 3-layer GPT-2 而后 finetune 等价于将原本的 full ckpt 进行压缩；它们的特征提取能力可以得到很好的保留，也符合正常的语义分析流程。

而后，对于 first 与 last 策略，我并没能有很好的分析方式。前者会丢失高维语义信息理解能力，后者会丢失底层信息的抽取能力，并不能直接做出预判，然而效果均不如 skip 策略是符合预期的。

Num Headers

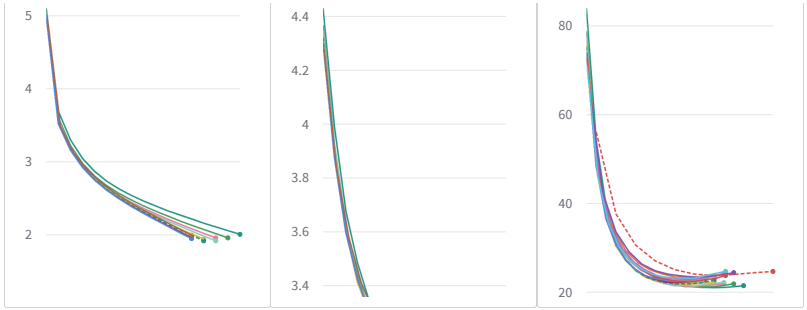
我将 header number 进行分解，遍历了所有可行的 header num。设计的实验如下：

```
experiments = [(1, 128), (768, 64), (2, 128), (384, 96), (3, 128), \
(256, 106), (4, 128), (192, 110), (6, 128), (128, 118), (8, 128), \
(96, 120), (12, 128), (64, 122), (16, 128), (48, 124), (24, 128), (32, 126)]
# num_header batch_size
 layer = 3
```

完全对于 3 层的 scratch 模型进行训练，其他参数均为默认参数值。

实验结果如下：

train_loss	val_loss	val_ppl



实际上，能够看出，num_header 对于模型的表现力影响并不显著（因为图片的图例较小，才能够看出差距，实际上影响是并不大的），即便是 num_header = 1，其 val_ppl 也只有 21.477，可以体现出 transformer 对于 num header 参数并不尽然敏感；然而，当 num_header 从 12 之后持续增加时，val_ppl 会有一定程度上升，可能是因为不同 header 间能力重叠严重，每个 header 都不够强大，一定程度降低了表现力。不过，对于 num_header 的不敏感可能是因为下游任务还是太简单了些，没有体现出 multi-header 的强大之处。

最后，过于多的 multi-head 而不配合增大 hidden_dim 可能会降低表现力，不过增大 num_header 能够加快模型的收敛速度，可能是因为并行 header 能够加速对于特征的捕捉过程。