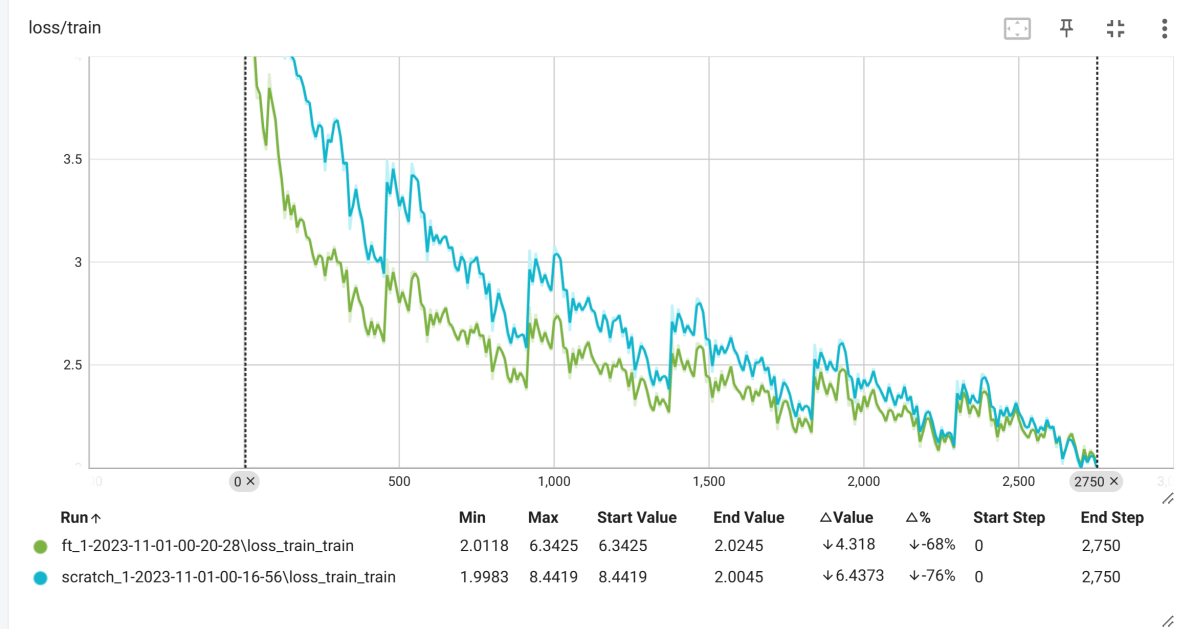
HW3-Report

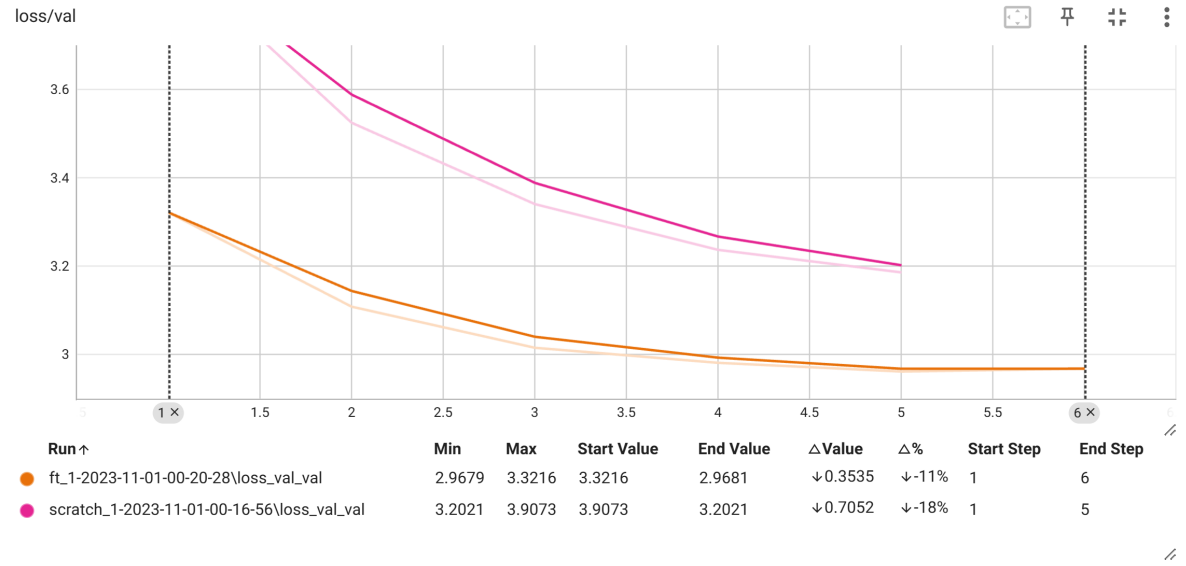
计11班 周韧平 2021010699

Tfmr-scratch && Tfmr-finetune 训练效果

训练过程中在 train 集合和 validation 集合上的 loss 曲线如下，训练时，我采用了框架默认的早停技巧，即在perplexity 到达最低点后停止训练



Run	Min	Max	Start Value	End Value	△Value	△%	Start Step	End Step
ft_1-2023-11-01-00-20-28\loss_train_train	2.0118	6.3425	6.3425	2.0245	↓4.318	↓-68%	0	2,750
scratch_1-2023-11-01-00-16-56\loss_train_train	1.9983	8.4419	8.4419	2.0045	↓6.4373	↓-76%	0	2,750



Run	Min	Max	Start Value	End Value	△Value	△%	Start Step	End Step
ft_1-2023-11-01-00-20-28\loss_val_val	2.9679	3.3216	3.3216	2.9681	↓0.3535	↓-11%	1	6
scratch_1-2023-11-01-00-16-56\loss_val_val	3.2021	3.9073	3.9073	3.2021	↓0.7052	↓-18%	1	5

选取验证集上最好的模型，他们在默认解码策略 （random, $\tau = 1$）下在测试集上的各项指标如下

Model	Perplexity	Forward BLEU	Backward BLEU	Harmonic BLEU
Tfmr-scratch	19.08	0.580	0.432	0.495
Tfmr-finetune	15.63	0.573	0.433	0.493

实验分析

- **微调模型相比随机初始化模型收敛速度更快，且在验证集和测试集上的 Perplexity 也更低**，这是因为模型从预训练模型中获得了更多的语言知识，这有助于其更高效地学习训练集上的语言分布，因此能够取得更快的收敛速度和更低的 ppl 指标
- **微调模型和随机初始化模型相比 BLUE Score 没有明显优势，甚至在 Forward BLEU 和 Harmonic BLEU 这两项指标上略低于后者**，考虑到ppl上微调模型具有明显优势，这一现象颇为反常。分析可能有如下原因：
    - PPL度量的是模型对其预测的不确定性。一个较低的PPL意味着模型在预测下一个词时更为自信。而BLEU评估的是模型生成文本与参考文本之间的 n-gram 重叠。这意味着一个模型可能在预测时更为自信，但这并不保证其生成的内容与参考文本更为接近。
    - BLEU 的计算公式只关心准确率，不关注召回率，这使得长度较短的式子在计算 BLUE 时往往更有优势，比如下面的例子

        Candidate: of the

        Reference 1: It is a guide to action that ensures that the military will forever heed Party commands.

        Reference 2: It is the guiding principle which guarantees the military forces always being under the command of the Party.

        Reference 3: It is the practical guide for the army always to heed the directions of the party.

    尽管论文中给出的 brevity penalty 策略可以在一定程度上起到惩罚召回率的作用，但在实践中大家发现这种惩罚往往是不够用的。而通过分析 Tfmr-scratch 和 Tfmr-finetune 两种策略生成的句子，**Tfmr-finetune 在 60% 的句子长度都严格大于随机初始化**，这可能是 pretrained 模型原有知识影响导致。而这一特性导致微调模型在BLEU指标上不如随机初始化

比较不同解码策略

Tfmr-scratch (Test Perplexity 19.08)

Strategy	temperature	Forward BLEU	Backward BLEU	Harmonic BLEU
random	$\tau = 1.0$	0.580	**0.432**	0.495
random	$\tau = 0.7$	0.827	0.384	**0.525**
top-p=0.9	$\tau = 1.0$	0.706	0.417	**0.525**
top-p=0.9	$\tau = 0.7$	**0.895**	0.304	0.453

Tfmr-finetune (Test Perplexity 15.63)

Strategy	temperature	Forward BLEU	Backward BLEU	Harmonic BLEU
random	$\tau = 1.0$	0.573	**0.433**	0.493
random	$\tau = 0.7$	0.809	0.387	**0.524**
top-p=0.9	$\tau = 1.0$	0.692	0.414	0.518
top-p=0.9	$\tau = 0.7$	**0.882**	0.316	0.465

结果分析

- **两种训练模型，top-p=0.9 下 Forward BLEU 得分都要更高，但在 Backward BLEU 得分上都略有降低，这表明生成结果的流畅性更好，但多样性有所下降。**这是因为 top-p 策略相比 random 策略，缩小了 token 选取的范围，排除了那些模型认为更不常见的结果，这有助于提升文字生成的准确率，但也在一定程度上限制了生成的多样性。
- **两种训练模型，$\tau = 0.7$ 相比 $\tau = 1.0$ 下 Forward BLEU 得分更高，但 Backward BLEU 得分更低，这表明模型生成结果的流畅性更好，但多样性有所下降。**根据计算式 $P(x_t|x_{<t}) = Softmax(Linear(h_{t-1})/\tau)$ “温度”的减小会导致模型预测的”熵“的减小，表现为token预测的概率分布更加尖锐，原本概率高的 token 更容易被选取，而概率低的 token 更不容易被选取，这有助于提高模型生成的流畅度，但也导致结果多样性下降。
- 整体来看 **top-p 策略和 τ 的选取本质上都是寻找模型生成的准确性和多样性之间的 trade-off**，通过观察综合性指标 Harmonic BLEU 结果我们可以得出以下结论：
    - **温度的选取取决于解码策略**，当我们采用 random 策略时，每个 token 都会被分配一定的选取概率，这时候模型生成的多样性好但流畅度不足，于是需要用较小的温度值让token的概率分布更加尖锐，从而提升生成的流畅度。而采用 top-p 策略时，由于策略本身已经排除了那些不太可能出现的token，为模型生成的流畅度提供了保障，因此选用较大的温度，可以保证模型的多样性不受影响
    - 两种模型在解码策略相同，温度相同的情况下结果也略有差异，对于 finetune，其在更为激进的 top-p=0.9 $\tau = 0.7$ 策略组合下表现要好于随机初始化模型，这也印证了前一节的分析，预训练的语言模型在大量数据上训练，可能已经学到了很多常见的语言模式。**当这种模型进行微调时，它可能会产生更为“平滑”或“一般化”的文本，这意味着它可能倾向于生成更为常见或不太特定的句子。对应到 token 的概率分布也更为平滑，因此选用更为激进的解码策略和更低的温度值，可以取得比随机初始化更好的结果。**

生成结果比较

下面是随机选取了10个位置，并将每组生成结果该位置下的句子提取出来的展示结果。我采用了word语法自动检查功能，为了避免标点符号被误判，我对于标点符号前的空格做了手动删除的操作。

	Scratch	Finetune
Random, t=1.0	A man is sitting on a bench while the cows. Three three giraffes facing each other in a fenced off by the wall. Three decorations on top of a bench and rail near the water. People walk on the same street between an American bus, stroll through the air. A stuffed bear a table near a rainbow children, sits by a tree. A picture of a city street at night with cars. A red bus at night in the middle of roadway. A blue bus race is parked in front of the protest. A mother giraffe standing next to two sheep in a field. White, blue and white photo of a woman sitting on a bench.	A man bus moves down the street while the sun sets. Three three giraffes facing each other in a fenced pasture. Three decorations on a bench beside a large rail on the hill. People walk on the same street between two buses and buses. A stuffed bear sticks to rub noses while children is standing by a tree. A picture of a city bus on street with cars and poles on the street. A red bus driving down the road near a roadway. A blue bus parked along a bike in a Asian city. A mother giraffe standing next to two adult giraffe in a pen. White, blue and blue transit bus placed between buildings.
Random, t=0.7	A man is sitting on a bench while the dog sits next to a bench. A man is sitting on a bench with two children on it. A man sits in a field with a dog on the ground. A giraffe standing in front of a tree ' s leaves. A stuffed bear is sitting on top of a bench. A red fire hydrant sitting on the side of the road. A red bus driving down the road past a roadway. A man sits on a bench in front of a wall. A mother giraffe standing next to a tree in a field. A giraffe eating off a tree in a forest.	A man is sitting on a bench while the sun is setting. A man is sitting on a bench with his dog on the leash. A man sits in a field with a rail on the ground. A giraffe standing in front of a group of trees. A pair of white buses parked in front of a red brick building. A man walks down the sidewalk on the street in the rain. A red bus driving down a street next to roadway. A bus full of vehicles driving past a bus stop. A couple of giraffes stand together in a field in a grassy field. A giraffe eating leaves from a tree branch in a field.
Top-p=0.9, t=1.0	A man is sitting on a bench while the cows. Three three giraffes facing each other in a fenced off area. Three people on a bench beside a large rail near the water. A giraffe standing in front of an adult fence near a tree. A stuffed bear a table on top of a wooden park bench. A picture of a city street at night with cars. A red bus at night in the middle of the road. A blue bus parked along a street in a city. A mother giraffe standing next to two sheep in a field. Two giraffes are standing around a tree in the grass.	A man is sitting on a bench while the sun is setting. Three three giraffes facing each other in a fenced pasture. Three people ride a bus on a large street near buildings. People walk on the same street between two buses and buses. A stuffed bear sticks to rub noses while other groups sits by a tree trunk. A picture of a city bus on street with cars and traffic lights. A red bus driving down a road near a tall building. A blue bus parked on the side of a street. A mother giraffe standing next to two adult giraffe in a pen.
Top-p=0.9, t=0.7	A man is sitting on a bench while the dog sits on a sidewalk. A man is sitting on a bench with two children on it. A man sits in a field with a dog on the ground. A giraffe standing in a field next to a tree. A very cute young child on a bench and some pigeons. A red fire hydrant sitting on the side of the road. A red bus driving down a street next to a traffic light. A man sits on a bench in front of a wall. A couple of giraffes stand together in a field in a grassy field. A giraffe eating off a tree in a forest.	A man is sitting on a bench while the sun is setting. A man is sitting on a bench with his dog on the leash. A man sits in a field with a horse on the side of the road. A giraffe standing in front of a fence with trees in background. A very cute young boy is looking at the sheep. A man is sitting on a bench with a dog. A red bus driving down a street next to a traffic light. A bus is travelling on a busy street with people walking in the background. A couple of giraffes stand together in a field. A giraffe eating leaves from a tree branch in a field.

- 在我选取的结果中，$\tau = 1.0$ 的组均存在语法错误，包括动词三单（other groups **sits**），重复出现数词（Three **three**），出现多个名词（**rainbow** children，**adult** giraffe），**有趣的是 $\tau = 0.7$ 的组在我选取的样本中没有出现语法错误，语言也更为流畅**，这也符合上一节中结果展示的$\tau = 0.7$ 的组具有更高的 Forward BLEU 值，两个事实共同印证了低温度下生成结果流畅度更高这一结论
- 从生成结果的多样性来看，**两种模型在 Random，t=1.0 组多样性最好**，具体来说，$\tau = 0.7$ 的组只有以冠词开头 "A [Noun]....."这样的句式，而 $\tau = 1.0$ 组会生成以数词开头的句子（Three people.....），而 Random，t=1.0 则会生成更多样化的名词及其修饰，例如 “White, blue and white photo of a woman sitting on a bench.” 和 White, blue and blue transit bus placed between buildings.“ 这样的句子就从没有在其它组别里出现过。这与上一节中这一组setting的Backward BLEU得分最高是一致的。

- 基于模型之间的横向比较，从阅读感受上来看 Finetune 模型的多样性更好，**Finetune模型具有更加丰富的词汇搭配。**例如，”city“ 一词的两组模型出现过的前后搭配包括 “a city bus ”，”a city street “,"a city"等，但Finetune 中特有 ”Asian city“ 这一搭配形式，这也是Asian这一词汇在结果中唯一一次出现。再如 Finetune 模型中出现的特有名词 ”leash“、”background“、” traffic lights“ 等，这些词均没有在 Scratch 模型中出现过。从句子长度来看，Finetune 模型倾向于用更长（有时甚至略显冗余）的的句子表述更丰富的含义（A picture of a city street at night with cars. vs. A picture of a city bus on street with cars and **poles on the street**），这从侧面印证了前两节中对预训练语言模型特性的解释，由于见过更多的文本数据，预训练语言模型在知识的丰富性上要优于随机初始化模型，也更倾向于生成不常见的表述。
- 综上所述，如果从个人阅读体验出发，我更倾向于选择在Random，t=1.0 组下的结果，因为在不影响理解的情况下，少量的语法错误可能对我来说并不会造成很大的干扰，而生成句子的多样性和合理性更是我首要考虑的因素。而在 Harmonic BLEU 指标下的最优的 Random， top-p=0.7 的结果，在看重流畅度和语法准确度的场景下则是最佳的选择，如果在考虑生成质量的稳定性的话会选取这个设定。

最终结果

最终我选择默认参数设定下Finetune模型在 Random， top-p=0.7 设定的结果，其它默认参数如下

```
seed = 1229
batch_size = 32
learning_rate = 1e-4
maxlen = 35
```

其在测试集上的定量化指标如下

Strategy	temperature	Perplexity	Forward BLEU	Backward BLEU	Harmonic BLEU
random	$\tau = 0.7$	15.63	0.809	0.387	0.524

结果保存在 `./output.txt` 中

Transformer Decoder 分析

Transformer vs. RNN

时间复杂度

- Transformer：Transformer 架构引入了自注意力机制，使其能够在输入序列的所有位置进行并行处理，而不需要像 RNN 那样顺序处理。这使得 Transformer 在训练和推理时具有更高的并行性，从而减少了时空复杂度。但由于自注意力矩阵的计算成本较高，Transformer 在较大的输入序列上可能会有较高的计算复杂度。
- RNN：RNN 是逐步处理输入序列的，因此在训练和推理时具有更低的并行性。此外，RNN 还受限于时间步的依赖关系，这可能导致较长的输入序列在训练时容易出现梯度消失或梯度爆炸问题。
- 具体来说，对Transformer，其自注意力层（self-attention layer）的时间复杂度为 $O(T^2d)$，其中 T 是序列长度，d 是输入向量维度。Feedforward 网络的时间复杂度为 $O(Td^2)$，而对于序列长度为 T 的 RNN，总的时间复杂度为 $O(Td^2)$，因此，在序列长度不大的情况下，二者的时间复杂度都等于 $O(Td^2)$，而在较大的序列长度下，Transformer block复杂度更高

性能

- 由于其并行性和能够捕捉长距离依赖关系的能力，Transformer 在各种自然语言处理任务（如机器翻译、文本生成等）中取得了显著的性能提升。它还可以通过使用更大的模型规模进一步提高性能，十分容易scale-up，例如 GPT-3 和 BERT。
- RNN 在某些序列建模任务中表现良好，尤其是对于时间序列数据。但对于自然语言处理任务，RNN 在处理长距离依赖关系时可能效果较差，因此通常需要使用更复杂的变种，如长短时记忆网络（LSTM）或门控循环单元（GRU）。

位置编码

- Transformer 使用位置编码来为输入序列的每个位置提供信息，以帮助模型捕捉序列中的顺序关系。通常使用三角函数的组合来生成位置编码，这样模型可以学习到不同位置之间的相对距离。
- RNN 自然地处理顺序信息，不需要显式的位置编码。但在某些情况下，可以将位置信息作为额外的输入特征提供给 RNN 模型。

推理时间复杂度

为什么设置 `use_cache = True`

在推理时为了加快速度，将每次 K,V 的计算结果保存下来，具体来说，为了求第 t 个token的输出，由于 K_{t-1} 和 V_{t-1} 在上一步已经算出来了，因此只需要计算第 t 个token 对应的 q，k，v 即可，即下面这行代码

```
query, key, value = self.c_attn(hidden_states).split(self.split_size, dim=2)
```

而通过将此前的 K_{t-1} 和当前的 k_t 以及此前的 V_{t-1} 和当前的 v_t 合并起来就可以得到 K_t 和 V_t，也就是下面的第一个 `if` 中的操作，同时将新的K_t 和 V_t 存储下来，为下一次运算使用，即第二个 `if` 中的操作

```
if layer_past is not None:
    past_key, past_value = layer_past
    key = torch.cat((past_key, key), dim=-2)
    value = torch.cat((past_value, value), dim=-2)

if use_cache is True:
    present = (key, value)
```

时间复杂度计算

对于`inference`中产生 l_t 的一次循环，其时间复杂度计算过程如下：

- 设定为 $hidden\_state = d$,feed forward layer 中间层维度为 $4d$，multi-head 头数为 n，transformer blocks 数量为 B，词表大小为 V
- 对于单层transformer block，生成 q, k, v 的时间复杂度为 $O(3d^2)$，multi head每个头计算qk乘积以及最后和v乘积的复杂度为 $(O(t * d/n))$，投影层时间复杂度为 $O(d^2)$，投射层由两层线性层构成，复杂度各为$O(4d^2)$，因此单层 transformer 的时间复杂度为 $O(3d^2 + n * t * d/n + d^2 + 2 * 4d^2) = O(d^2 + dt)$
- 从 transformer output 到词 ID的过程时间复杂度为 $O(dV)$，无论采用何种解码策略，其复杂度也不过是$O(d^2)$可以忽略，因此解码单个token的时间复杂度为 $O(B * (d^2 + dt) + dV) = O(Bd^2 + Bdt + dV)$

综上，根据上述分析，生成l_t的复杂度为 $O(Bd^2 + Bdt + dV)$，生成一个长度为 T 的完整序列，其复杂度为 $O(TBd^2 + BdT^2 + dVT)$

主导模块

根据上述分析，当序列长度 T 较大时，和序列长度有关的 self-attention 模块占据主导，而当 transformer 维度 d 较大时，贡献 $O(Bd^2)$ 项的 feed-forward 层占主导

预训练的影响

第一节中我们得出了结论：微调模型相比随机初始化模型收敛速度更快，且 Perplexity 也更低，但在 BLUE Score 上略弱于随机初始化的效果。这是受预训练模型的特点导致的，在一二三节中我们探讨了其原因：模型从预训练模型中获得了更多的语言知识，而这些 pretrained 模型原有知识并不保证其生成的内容与参考文本更为接近（第一节），甚至反而会使其生成更一般化、不寻常的结果（第二、三节），同时由于 Blue Score 计算方式决定其天然存在对一些文本特点的偏好，ppl 和 Blue score 对文本的要求也有所不同（第一节），导致微调模型表现不尽如人意。

Bonus

讨论 BPE tokenizer 的优势

BPE（Byte-Pair Encoding）分词相比传统的基于空格的分词具有的优势包括：

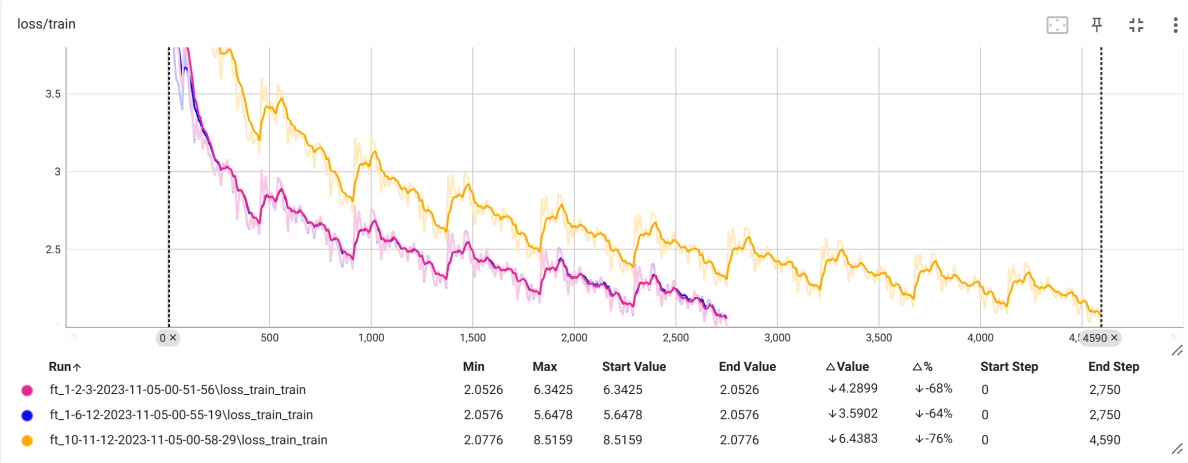
1. BPE分词可以将复杂的单词拆分为更小的部分，从而更好地处理复杂的语言结构。相比之下，基于空格的分词方法无法正确处理复合词或连字符的单词，限制了其适应性和准确性。同时，对于未知单词或者词组，BPE分词可以通过逐步拆分单词来处理。这使得BPE分词在处理未见过的单词和专业术语时更具优势。BPE分词不依赖事先建立的词典，因此可以更好地处理未见过的词组。相比之下，基于空格的分词方法需要事先将所有可能的词组列入词典中。
2. 减少数据噪声影响：在文本数据中，常常存在标点符号、大小写和拼写错误等噪声。BPE分词可以将单词拆分为子单词，从而减少噪声对模型的影响。而基于空格的分词方法无法有效处理或区分这些噪声。这样做可以有效提高模型泛化能力，因为它通过拆分单词为子单词，使得相似的词汇和语境能够共享子单词，从而更好地捕捉语义和上下文信息。

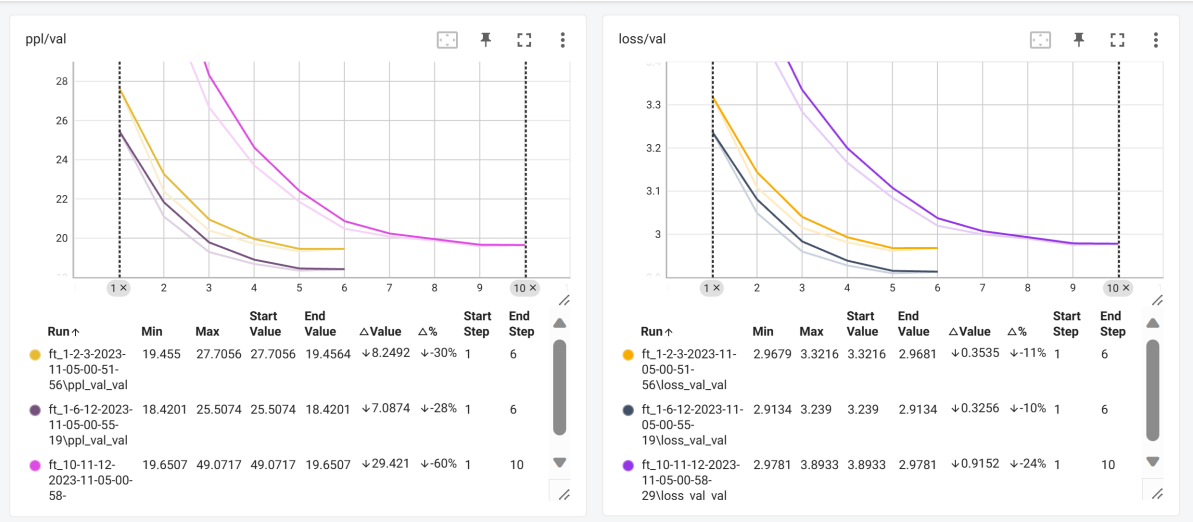
具体来说，在下面的例子中，BPE tokenizer 可以将 Traveling拆成 Travel 和 ing，将 disadvantageous 拆成 disadvantage 和 ous 等，这类操作可以将词汇的后缀和前缀去掉，使同一词义的不同形态有共同的token部分，同时也可以缩小词表的容量，降低空间需求，除此以外，像通过连字符连接的词汇 life-changing 也可以被拆成两个更为熟悉的词汇，可以提高模型对陌生词的理解能力

```
Traveling to a new country can be a life-changing experience. =====>
['Travel', 'ing', ' to', ' a', ' new', ' country', ' can', ' be', ' a', ' life', '-', 'changing', ' experience', '.']
The new policy had a disadvantageous impact on small businesses. =====>
['The', ' new', ' policy', ' had', ' a', ' disadvantage', 'ous', ' impact', ' on', ' small', ' businesses', '.']
After careful reconsideration, they decided to change their plans. =====>
['After', ' careful', ' reconsider', 'ation', ',', ' they', ' decided', ' to', ' change', ' their', ' plans', '.']
Decentralization allows for greater autonomy and decision-making power at the local level. =====>
['Dec', 'ent', 'ral', 'ization', ' allows', ' for', ' greater', ' autonomy', ' and', ' decision', '-', 'making', ' power', ' at', ' the', ' local', ' level', '.']
```

选用不同 transformer 层对模型训练效果的影响

本节中，我从GPT-2中的12层transformer中选取了不同的三层（前三层，后三层和1，6，12三层），在和此前相同的超参数下训练，并采用 random+$\tau = 0.7$ 的解码策略，得到了如下结果





在 Test 集合上的定量化结果如下：

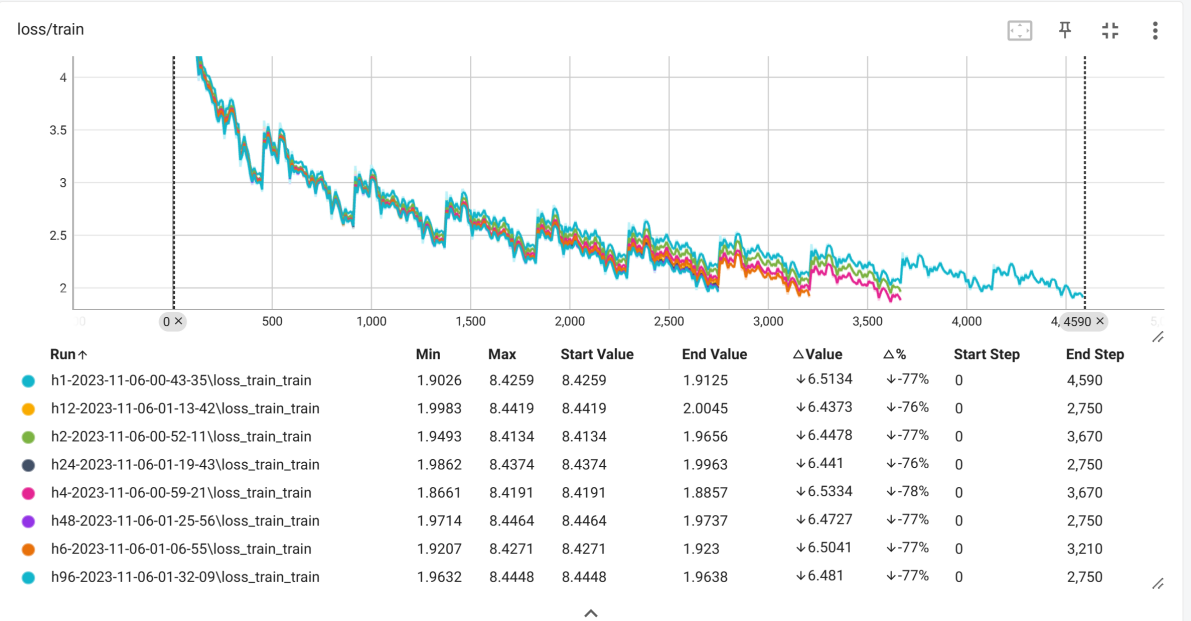
Transformer layers	Perplexity	Forward BLEU	Backward BLEU	Harmonic BLEU
1-2-3	15.63	0.809	0.387	0.524
1-6-12	**14.91**	**0.823**	0.389	**0.528**
10-11-12	16.16	0.812	**0.391**	**0.528**

实验分析

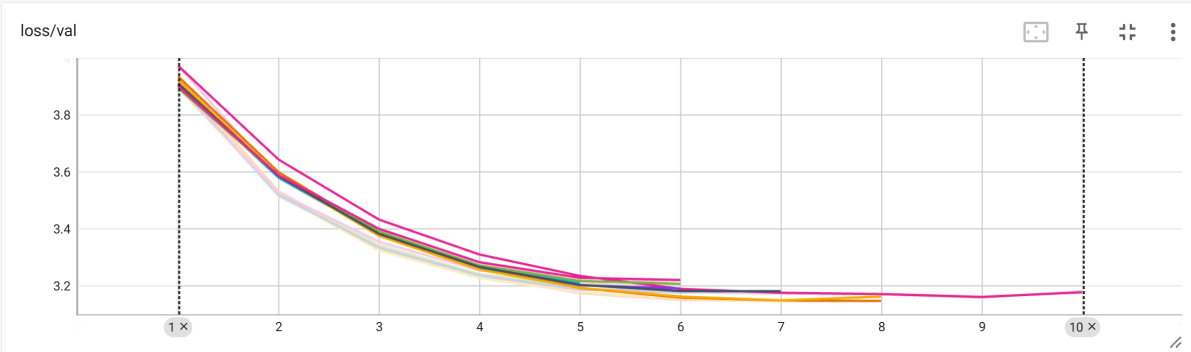
- 结果表明，和选前三层和选后三层这两种策略相比，跳跃选取1-6-12层的策略在validation集合上收敛最快且loss和ppl都最低，且在 Test 集合上ppl，Forward BLEU 和 Harmonic BLEU 也都是最好的。**这说明不同层的transformer block在预训练时学习的侧重有所不同**，例如靠前的层可能更倾向于学习局部、细节的特征，而靠后的层则倾向于学习更加宏观、整体的特征，因此**跳跃选取transformer层相比连续选取是将12层模型压缩到三层的更好方式**
- **ppl 指标并不总能指示模型生成质量**，如前文讨论，ppl只能说明模型在生成文字时的自信程度，但这种较高地置信度并不等价于更高的生成质量，因此10-11-12层模型在 Test perplexity 上最大，却反而能表现出很好的Harmonic BLEU 得分

讨论 multi-head attention 中 head 数量的影响

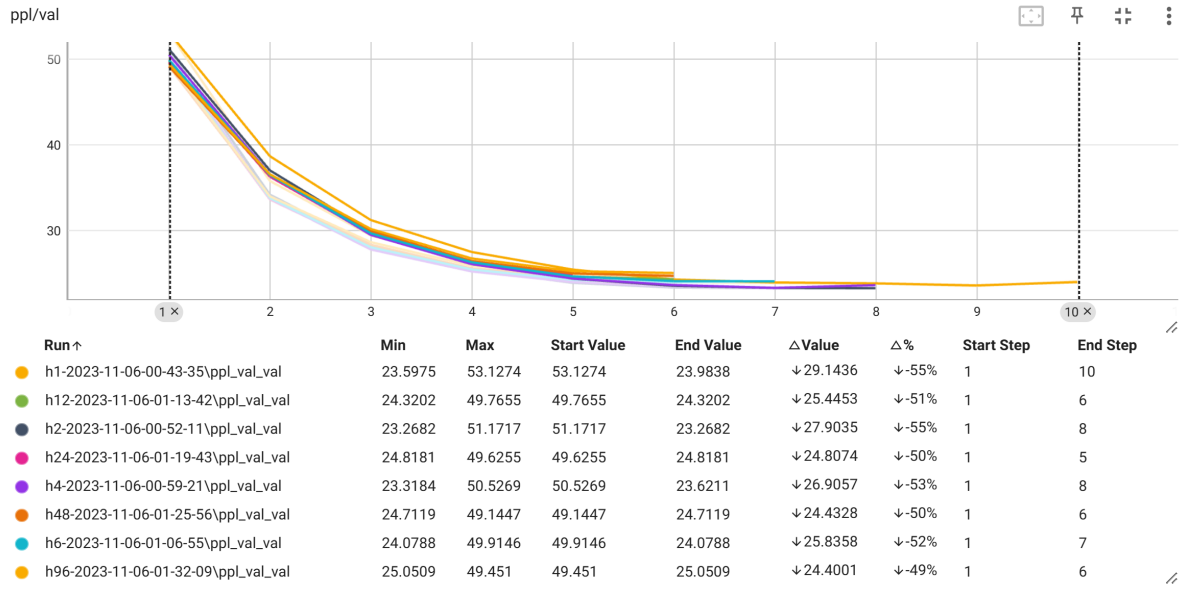
本节中，我采用 random+$\tau = 0.7$ 的解码策略，分别在 head num= 1，2，4，8，12，24，48，96 下对随机初始化的模型进行训练，得到如下结果



Run	Min	Max	Start Value	End Value	△Value	△%	Start Step	End Step
h1-2023-11-06-00-43-35\loss_train_train	1.9026	8.4259	8.4259	1.9125	↓6.5134	↓-77%	0	4,590
h12-2023-11-06-01-13-42\loss_train_train	1.9983	8.4419	8.4419	2.0045	↓6.4373	↓-76%	0	2,750
h2-2023-11-06-00-52-11\loss_train_train	1.9493	8.4134	8.4134	1.9656	↓6.4478	↓-77%	0	3,670
h24-2023-11-06-01-19-43\loss_train_train	1.9862	8.4374	8.4374	1.9963	↓6.441	↓-76%	0	2,750
h4-2023-11-06-00-59-21\loss_train_train	1.8661	8.4191	8.4191	1.8857	↓6.5334	↓-78%	0	3,670
h48-2023-11-06-01-25-56\loss_train_train	1.9714	8.4464	8.4464	1.9737	↓6.4727	↓-77%	0	2,750
h6-2023-11-06-01-06-55\loss_train_train	1.9207	8.4271	8.4271	1.923	↓6.5041	↓-77%	0	3,210
h96-2023-11-06-01-32-09\loss_train_train	1.9632	8.4448	8.4448	1.9638	↓6.481	↓-77%	0	2,750



Run	Min	Max	Start Value	End Value	△Value	△%	Start Step	End Step
h1-2023-11-06-00-43-35\loss_val_val	3.1611	3.9727	3.9727	3.1773	↓0.7954	↓-20%	1	10
h12-2023-11-06-01-13-42\loss_val_val	3.191	3.9073	3.9073	3.191	↓0.7163	↓-18%	1	6
h2-2023-11-06-00-52-11\loss_val_val	3.1471	3.9352	3.9352	3.1471	↓0.7881	↓-20%	1	8
h24-2023-11-06-01-19-43\loss_val_val	3.2105	3.9045	3.9045	3.2105	↓0.694	↓-18%	1	5
h4-2023-11-06-00-59-21\loss_val_val	3.1491	3.9225	3.9225	3.1621	↓0.7604	↓-19%	1	8
h48-2023-11-06-01-25-56\loss_val_val	3.207	3.8948	3.8948	3.207	↓0.6877	↓-18%	1	6
h6-2023-11-06-01-06-55\loss_val_val	3.1813	3.9103	3.9103	3.1813	↓0.729	↓-19%	1	7
h96-2023-11-06-01-32-09\loss_val_val	3.2207	3.901	3.901	3.2207	↓0.6803	↓-17%	1	6



Run	Min	Max	Start Value	End Value	△Value	△%	Start Step	End Step
h1-2023-11-06-00-43-35\ppl_val_val	23.5975	53.1274	53.1274	23.9838	↓29.1436	↓-55%	1	10
h12-2023-11-06-01-13-42\ppl_val_val	24.3202	49.7655	49.7655	24.3202	↓25.4453	↓-51%	1	6
h2-2023-11-06-00-52-11\ppl_val_val	23.2682	51.1717	51.1717	23.2682	↓27.9035	↓-55%	1	8
h24-2023-11-06-01-19-43\ppl_val_val	24.8181	49.6255	49.6255	24.8181	↓24.8074	↓-50%	1	5
h4-2023-11-06-00-59-21\ppl_val_val	23.3184	50.5269	50.5269	23.6211	↓26.9057	↓-53%	1	8
h48-2023-11-06-01-25-56\ppl_val_val	24.7119	49.1447	49.1447	24.7119	↓24.4328	↓-50%	1	6
h6-2023-11-06-01-06-55\ppl_val_val	24.0788	49.9146	49.9146	24.0788	↓25.8358	↓-52%	1	7
h96-2023-11-06-01-32-09\ppl_val_val	25.0509	49.451	49.451	25.0509	↓24.4001	↓-49%	1	6

head num	Perplexity	Forward BLEU	Backward BLEU	Harmonic BLEU
1	18.78	0.809	**0.389**	**0.525**
2	18.51	0.819	0.385	0.524
4	**18.50**	0.819	0.387	**0.525**
6	18.86	0.825	0.380	0.520
12	19.08	**0.827**	0.384	**0.525**
24	19.28	0.820	0.386	**0.525**
48	19.42	0.825	0.385	**0.525**
96	19.53	0.819	0.382	0.521

实验分析

- **训练中，随着模型 attention head 数量的增加，模型训练的收敛速度是逐步加快的。**这是因为通过使用多头注意力，模型可以生成多个独立的特征表示，这些表示在不同的头部中具有不同的权重。这种多样性的特征表示能够更好地捕捉输入序列中的细微差异和变化，从而提高模型的表达能力和泛化能力。同时，在多头注意力模型中，每个头部都有自己的参数集合，这意味着每个头部都有独立的梯度。通过将不同头部的梯度进行平均或合并，可以帮助更好地传播梯度信号，使得梯度信息更加充分地传递到模型的各个层次，从而加速收敛过程。
- **模型在测试集上的 ppl 和 Forward BLEU 指标都呈现先变好再变差的趋势。**在 attention head 较少时，多头注意力模型可以从多个不同的视角同时获取信息，并在不同的注意力头部中进行交互。每个头部可以专注于不同的特征子空间，从而捕捉到更多的上下文信息。这种信息的交互可以帮助模型更好地理解输入序列的全局依赖关系，进而提高模型的泛化能力。而随着 attention head 增多，每个头内包含的维度更少，不容易学到有用的信息，导致学习效果的下降，还可能存在过拟合的问题，导致模型 ppl 和 Forward BLEU 指标下降
- **模型在综合性指标 Harmonic BLEU 上，不同头数量的表现没有显著差异。**这可能是收到任务本身和语料库大小的限制，当训练集本身不大时，多 head 相比单 head 并不能提取到更多的有用信息，模型生成能力也不会有显著提高

参考资料

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).