HW8 Report

學號：b07901039　系級：電機二　　姓名：劉知穎

1. (20%) Teacher Forcing:
   a. 請嘗試移除 Teacher Forcing，並分析結果。

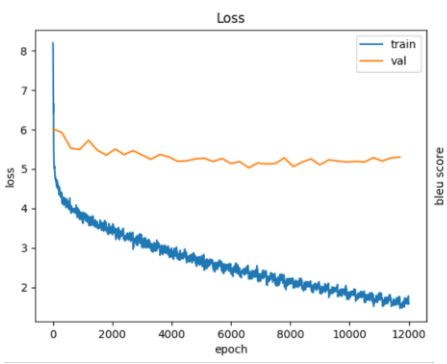
      Seq-to-seq 模型架構

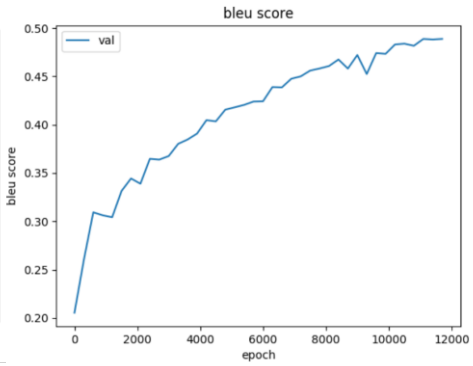
```
Seq2Seq(
  (encoder): Encoder(
    (embedding): Embedding(3922, 256)
    (rnn): GRU(256, 512, num_layers=3, batch_first=True, dropout=0.5, bidirectional=True)
    (dropout): Dropout(p=0.5, inplace=False)
  )
  (decoder): Decoder(
    (embedding): Embedding(3805, 256)
    (attention): Attention()
    (rnn): GRU(256, 1024, num_layers=3, batch_first=True, dropout=0.5)
    (embedding2vocab1): Linear(in_features=1024, out_features=2048, bias=True)
    (embedding2vocab2): Linear(in_features=2048, out_features=4096, bias=True)
    (embedding2vocab3): Linear(in_features=4096, out_features=3805, bias=True)
    (dropout): Dropout(p=0.5, inplace=False)
  )
)
```

      batch size = 60、epoch = 12000、每 300 epoch 做 validation 避免 overfitting。

      使用 Adam optimizer、learning rate = 5e-5。

      (1) 使用 teacher forcing



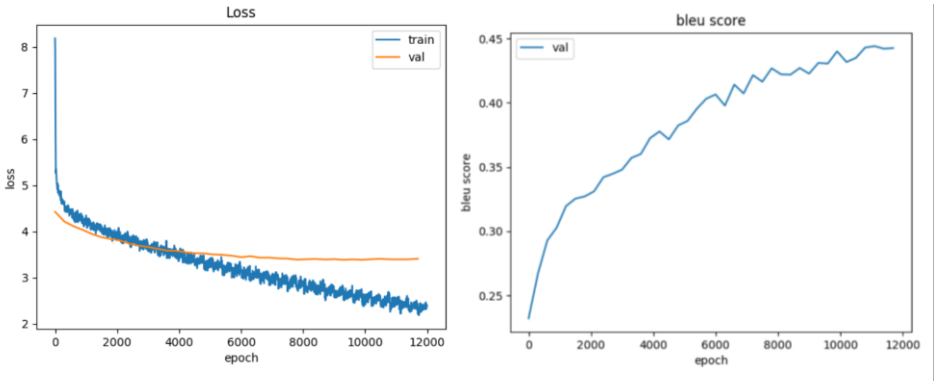


      實作方式：在 schedule_sampling 回傳 1。

      best validation loss: 5.300, bleu score: 0.487

      (2) 移除 teacher forcing

實作方式：在 schedule_sampling 回傳 0。



best validation loss: 3.406, bleu score: 0.443

結論：

完全移除 teacher forcing 而使用自己預測的單字當下一級 input 的 bleu score 較全用 teacher forcing 差。推論原因是使用自己預測出的單字在 training 時，當預測的單字變化時，gradient 的傳遞會有困難，因此訓練上較難訓練好。移除 teahcer forcing 後 loss 降低，推測原因是 traning 時 optimizer 的目標是降低 loss，且此時 training 和 testing 的過程 match，所以 validation 的 loss 降低。不過，通常以 bleu score 當作模型學習好壞的判斷標準，因此全用 teacher forcing 的結果是較好的。

2. (30%) Attention Mechanism:
   a. 請詳細說明實做 attention mechanism 的計算方式，並分析結果。

   (1) 無 attention mechanism

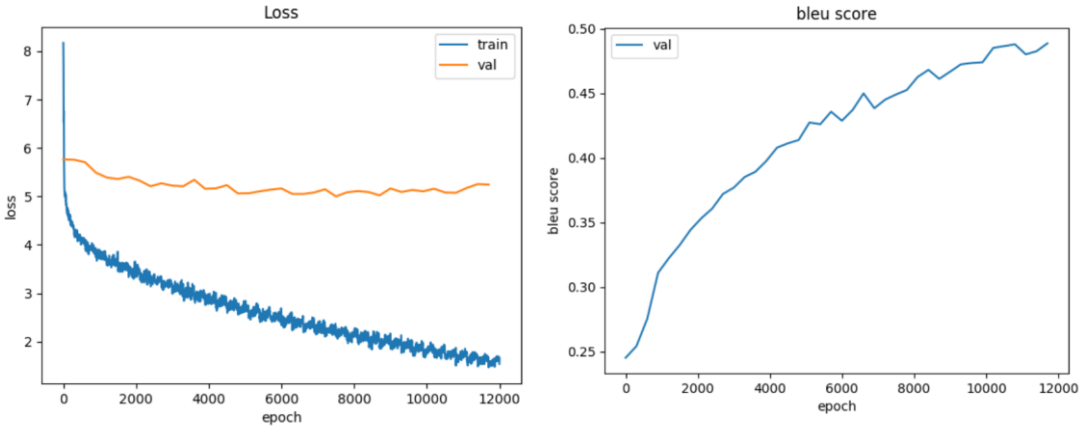
   模型架構和訓練方法和過程如第一題第一部分所描述(用 teacher forcing)。

   Best validation loss: 5.300, bleu score: 0.487

   (2) Content Base: Cosine Similarity (hidden 最後一層 layer)

   實作方式：

取 decoder hidden 的最後一層 layer 和 encoder output 做 cosine similarity。兩兩長度為 hidden dimension 的 vector，共 sequence length * batch size 個，得到 matrix alpha。接著讓 alpha 在 sequence length dimension 做 soft max。讓 soft max 後的 alpha 和 encoder output 做 element wise multiplication，然後再在 sequence length 方向



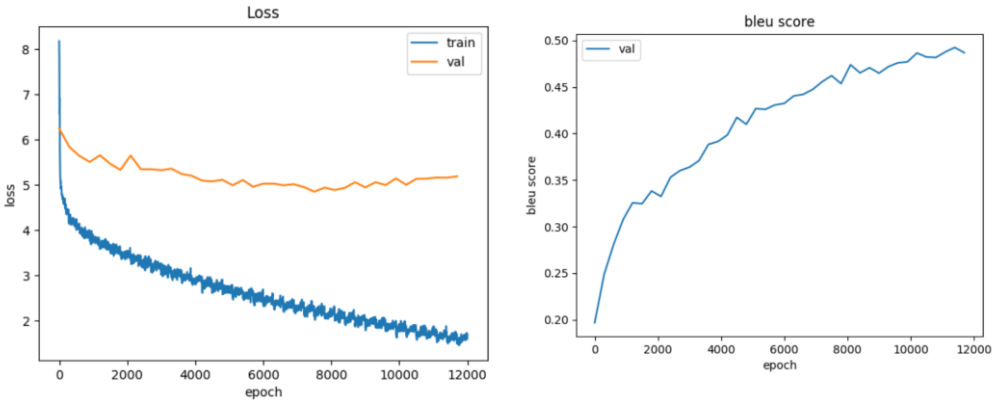
加起來。將得到的向量([batch size, 1, hid dim])回傳和 embeded vector 相連作為 GRU 的下一級的 input。

Best validation loss: 5.243, bleu score: 0.489

(3) Content Base: Cosine Similarity (hidden 第一層 layer)

實作方式：

和前一小題相同，唯改取第一層 hidden layer。(參考 Google’ s Neural Machine Translation 取第一層 hidden layer)



Best validation loss: 5.159, bleu score: 0.492

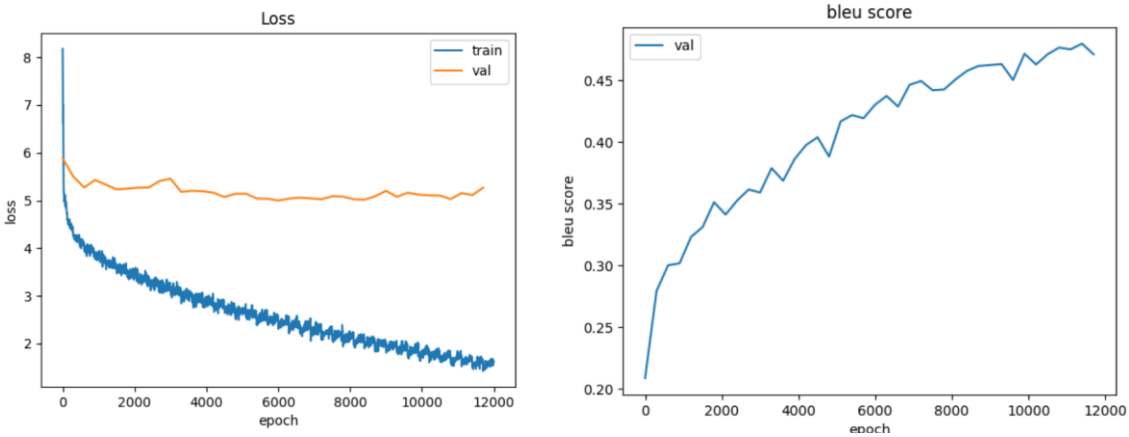
(4) Content Base: Cosine Similarity (hidden 最後一層 layer) + feed forward network 實作方式：

前置作業和第二小題相同。將得到的 attention vector 和 embedded vector 相連接，通過一個 feed forward network 後再輸入 GRU。(參考 Luong et. al (2015))

feed foward network 架構：

linear(input_dim , floor(input_dim/2)) / dropout(0.5) / relu() / linear(floor(input_dim/2), input_dim)

input_dim = emb dim + hid dim * 2

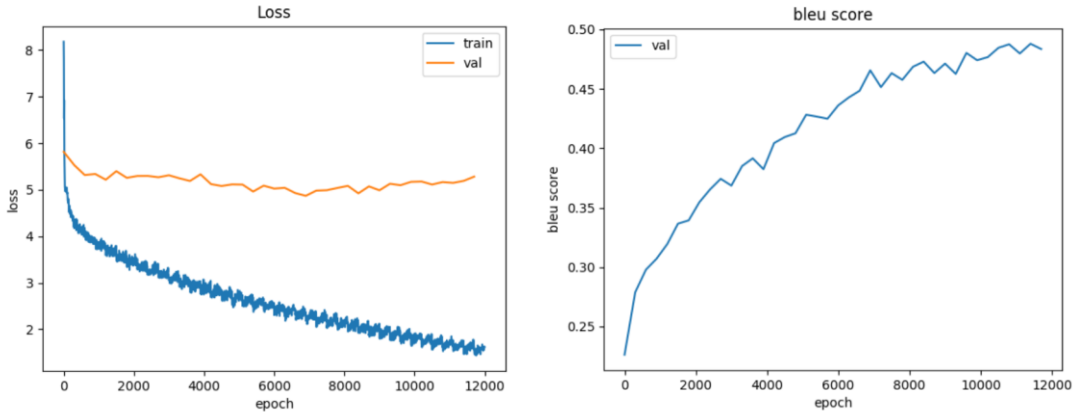


Best validation loss: 5.111, blue score: 0.480

(5) Content Base: Cosine Similarity (hidden 第一層 layer) + feed forward network

實作方式：

和前一小題相同，唯改取 hidden 的第一層 layer。



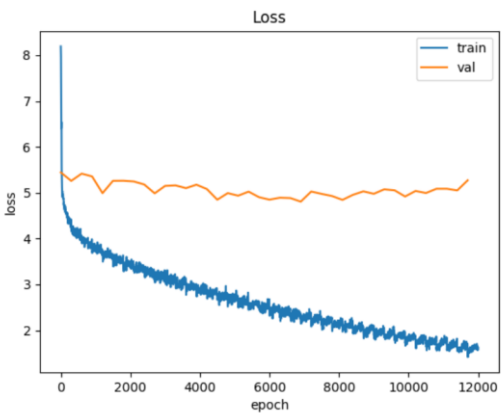
Best validation loss: 5.190, blue score: 0.488

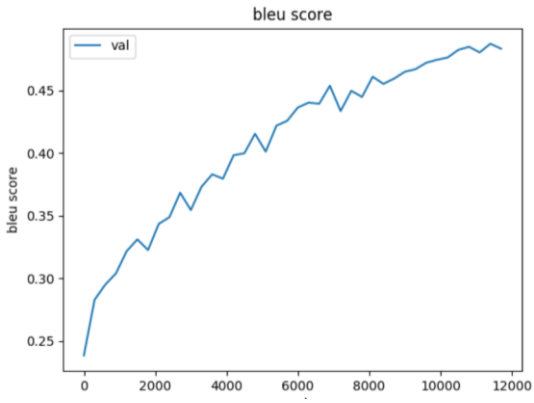
(6) Bahdanau Attention

$$score_{alignment} = W_{combined} \cdot tanh(W_{decoder} \cdot H_{decoder} + W_{encoder} \cdot H_{encoder})$$

實作方法：

將 decoder hidden 的第一層 layer 和 encoder output 分別經過 linear layer(hidden dim , hidden dim)後相加(pytorch 自動將 decoder hidden[0]展開成和 encoder output 相同維度)，通過 tanh。接著乘上一個 learnable matrix、經過 soft max 得到 attention weights。將 attention weights 和 encoder outputs 做 element wise 相乘，最後在





sequence length 維度相加得到 context vector。將 context vector 和 embeded vector 相連接，輸入 GRU。

Best validation loss: 5.051, bleu score: 0.487

結論：

(1) 在查詢資料時發現有些模型是取 decoder hidden 第一層 layer，而有些是取最後一層 layer 去和 encoder output 計算 attention。在這次實驗中，取第一層 layer 的效果較好。

(2) 有查到一些模型會將 context vector 和 embedded input 相連接的向量先通過 feed forward network 再輸入 GRU。但實作的效果不好，第四、五小題的結果較二、三小題差。然而，不排除是 feed forward network 的架構設計不好造成結果退步。

(3) 這次實驗中效果最好的是第三個方法—將 decoder hidden 的第一層 layer 的向量和 encoder output 計算 consine similarity。bleu score 從原本 0.487 進步到 0.492。

attention 原理參考：
https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3
第六小題實作參考：
https://blog.floydhub.com/attention-mechanism/

3. (30%) Beam Search:
   a. 請詳細說明實做 beam search 的方法及參數設定，並分析結果。

模型架構和訓練方法和過程如第一題第一部分所描述(用 teacher forcing、無 attention)。載入第一題訓練好的模型，在 validation 用不同 beam size 作 testing。

實作方式：

以 class node 存取每個節點的 input hidden vector、decoder input vector、previous node、log probability。用一個 list 紀錄要處理的 node。從第一個 input <BOS>建立第一個 root node (previous node = None、log probability = 0)開始，每次處理同一層在 list 裡的 parent nodes，將每個 parent nodes 前 beam size 機率大的 child nodes 放入 list 中。child node 的 log probability 是其 parent nodes 的 log probability 加上它的 output vector 在那個字上的數值取 soft max log 或取 log。每次處理完一個 node 後，就把它從 list 中去除。處理完同一層 parent nodes 後，此時 list 裡應該只有他們的 child nodes，留下前 beam size 機率大的 child nodes，當作下一級的 parent nodes。最後從最終的 node 用 previous node back trace。依照 back trace path 上的 nodes 計算 output vector，prediction vector 則直接用下一級 node 的 input 相連接。

無 beam search：

best validation loss: 5.300, bleu score: 0.487

(1) output probability 取 log soft max

Beam size	validation loss	bleu score
3	5.368	0.490
4	5.411	0.486
5	5.397,	0.487
7	5.398	0.491
10	5.390	0.486
15	5.379	0.487

(2) output probability 取 log

Beam size	validation loss	bleu score
3	5.493	0.477
4	5.511	0.475
5	5.592	0.474
7	5.600	0.470
10	5.695	0.469
15	5.764	0.457

結論：

(1) 將 output 的 probability 取 log soft max 得到的結果叫只取 log 好，推測是因為只取 log 時，可能被數值極端大的某一分支影響，而無法探索到其他分支。

(2) 使用不同 beam size 的 beam search(用 log soft max)雖然大致上能讓 bleu score 進步，但卻不是 beam size 越大，bleu score 就越高。

推測是因為 beam search 是去最大化機率，而不是去選 bleu score 較高的句子，且 bleu score 本身在計算上也有些瑕疵。

4. (20%) Schedule Sampling:

a. 請至少實做 3 種 schedule sampling 的函數，並分析結果。

模型架構和訓練方法和過程如第一題第一部分所描述(沒有用 attention)

實作方式：

ratio_epoch = current_epoch / total_epoch
schedule_sampling 根據 ratio_epoch 回傳使用 teacher forcing 的機率。和隨機生成 0 到 1 的數字相比，若 schedule_sampling 回傳的值較大就用 target 當下一級 input，反之用預測的 output 當下一級 input。

(1) Teacher forcing

schedule_sampling return 1

Best validation loss: 5.300, bleu score: 0.487

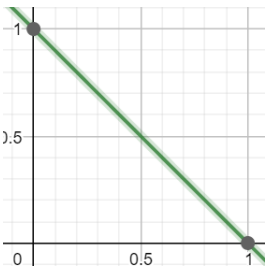
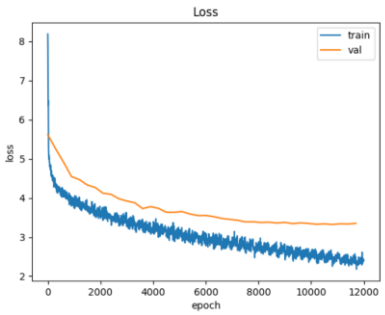
(2) No teahcer forcing

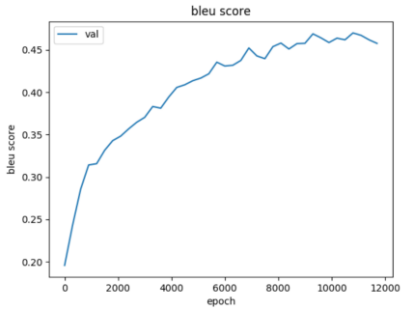
schedule_sampling return 0

Best validation loss: 3.406, bleu score: 0.443

(3) linear

schedule_sampling return (1 – ratio_epoch)

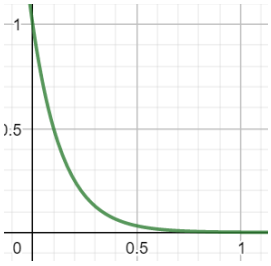
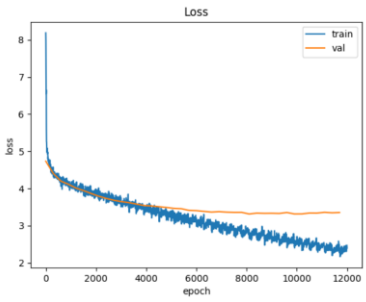



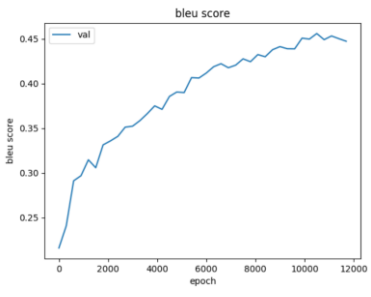


Best validation loss: 3.321, blue score: 0.470

(4) exponential decade

schedule_sampling return (1/1000)^ratio_step

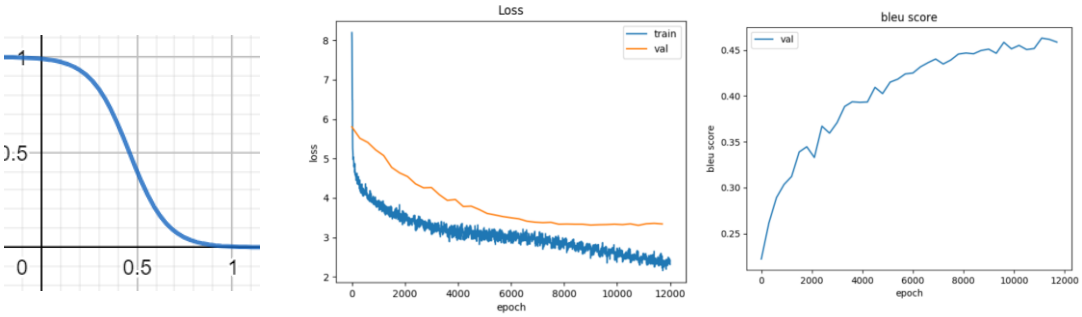


Best validation loss: 3.338, blue score: 0.456

(5) inverse sigmoid (k=1000, n=10)

$$f(x) = \frac{k}{k + e^{nx}}$$

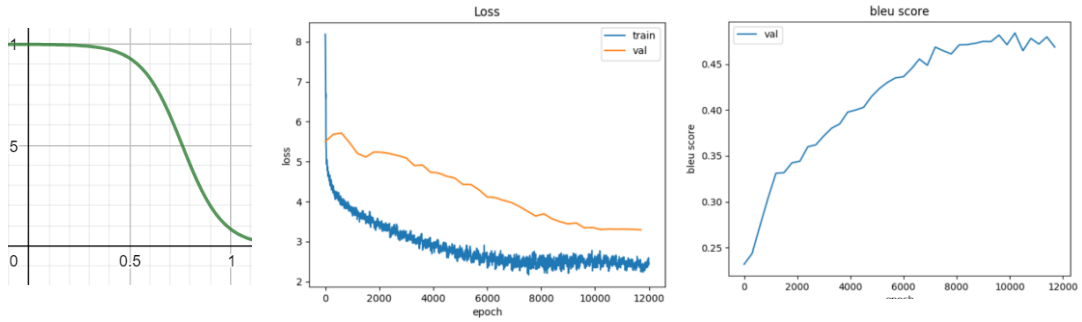
schedule_sampling return f(ratio_epoch)



Best validation loss: 3.356, blue score: 0.462

(6) inverse sigmoid (k=2000, n=10)

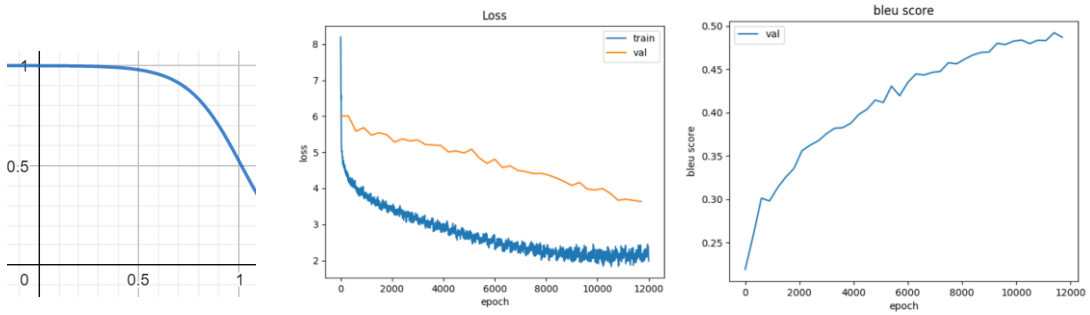
schedule_sampling return f(ratio_epoch)



Best validation loss:3.304, blue score: 0.484

(7) inverse sigmoid(k=2000, n=7.5)

schedule_sampling return f(ratio_epoch)



Best validation loss: 3.666, blue score: 0.492

結論：

(1) 第三到第六的 schedule sampling 後的結果都較原本全用 teacher forcing 差，只有最後一個 schedule sampling 的結果有稍微進步。

(2) 三到七的 schedule sampling 結果都較完全不用 teacher forcing 好，由此可知 teacher forcing 的重要性。即使是 exponential 的 schedule sampling 函

數，使用 teacher forcing 的機率很快衰減到 1，但只要有在前期訓練過程有較大機率用 teacher forcing 就會比完全沒用 teacher forcing 的結果好。

(3) 比較第五到七的函數，三個函數都是 inverse sigmoid 但有不一樣的衰減速度。從五到七，從一衰減到零的速度漸慢，而 bleu score 則漸大。由此可知，在某個零界點之前，使用越大比例的 teacher forcing 能讓結果變好。