學號：**B06902125** 系級：資工三 姓名：黃柏瑋

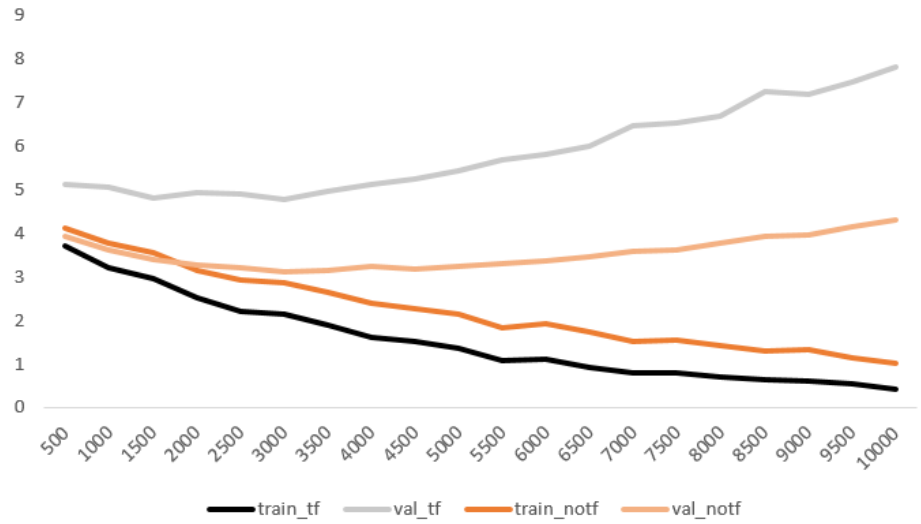
1. (20%) Teacher Forcing

請嘗試移除 **Teacher Forcing**，並分析結果。

首先，先簡單說明模型架構：原則上和助教提供的sample model一樣，但在decoder的RNN之前加了一個PReLU，並在classifier上多加一些dropout，避免overfitting太早發生。此外，為了加速訓練，將learning rate提升至2e-4，同時也將batch size提升至96，以減緩訓練時的過度震盪。

當訓練過程有teacher forcing與沒有teacher forcing時，learning curve和blue score的比較如下：

(train與val的crossentropy loss)



(val的BLEU)

	TF	NO TF
Valid Bleu	0.5831	0.5643
Test Bleu	0.5626	0.5416

由實驗結果可知，有用teacher forcing的training crossentropy loss比較低，代表teacher forcing可以使收斂更加有效率。而雖然valid crossentropy loss比較高，但無論valid或testing的BLEU score都比沒有用teacher forcing的多出2%，這也證實了teacher forcing可以有效提升seq2seq的訓練，以正確答案當作輸入能夠穩定模型參數模型的更新，進而達到較好的預測。

此外，從上面的結果也能發現crossentropy loss變差並不保證BLEU也會變差，甚至還可能使BLEU變好，畢竟crossentropy要求字對位置也對，而BLEU可以容許字對位置不對。由於主要的evaluation matrix為BLEU score，且testing dataset比valid dataset完整，故將選擇testing dataset的BLEU score作為評估模型的主要依據。

2. (30%) Attention Mechanism

請詳細說明實做 **attention mechanism** 的計算方式，並分析結果。

Attention的實作方式有很多種，可以用不同的方式算出attention weight，也可以將attention應用在decoder的不同位置上。

在本題中，比較以下三種不同的狀況：

1. 在過decoder RNN之前，取出encoder outputs與當前decoder RNN最後一層的hidden states，算出context vector之後，與原先的decoder input相接，放入RNN中繼續完成decoding。
2. 在過完decoder RNN之後，取出encoder outputs與當前decoder RNN的output(即最後一層的hidden states)，算出context vector之後，與原先的RNN output相接，放入classifier中繼續完成decoding。
3. 在過完decoder RNN之後，取出encoder outputs與當前decoder RNN所有層的hidden states，算出context vector之後，與原先的RNN output相接，放入classifier中繼續完成decoding。

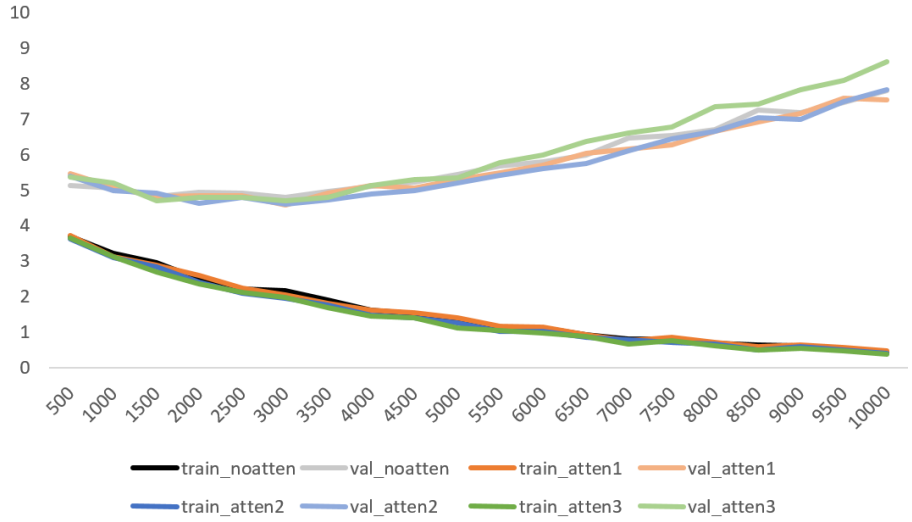
其中，context vector的算法如下，主要參考bahdanau attention，但省去在softmax之前取exponential的部分，因為實作上發現拿掉後結果較好。算式中，A為encoder outputs，B為decoder RNN hidden states或output，而W_1、W_2和V都是可被學習的linear weights：

$$atten\_weight = softmax(V(tanh(W_1(A) + W_2(B)))$$

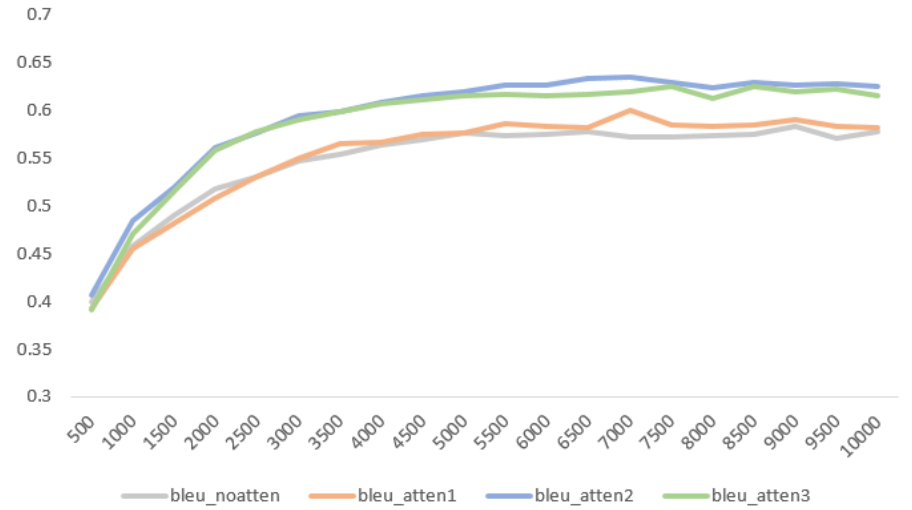
$$context\_vector = atten\_weight \cdot A$$

以下為learning curve及BLEU score的比較，基礎模型為第一題中的teacher forcing model：

(train與val的crossentropy loss)



(val的BLEU)



	NO ATTEN	ATTEN1	ATTEN2	ATTEN3
Test BLEU	0.5626	0.5618	0.6073	0.6026

首先，先觀察第一種attention與沒有attention的結果，會發現他們無論training loss、validation loss、validation BLEU或testing BLEU都相當接近，感覺這樣的接法並沒有產生任何效果，於是改接第二種attention看看如何。

第二種attention的效果比第一種顯著許多，在BLEU score上有4～5%的進步，代表在這個模型上，將attention接在RNN之後會比接在RNN之前更吃香。而我認為可能的原因如下：

1. decoder RNN的output(即最後一層hidden states)會被接入classifier，因此可以說是字的feature。第一種attention用了前一個字的feature計算attention weight，而第二種attention用當前的feature去算，感覺上更貼近該字的需求，導致結果較精準。
2. RNN主要是用來提取字的feature，而classifier才是真正的分類工具，因此將context vector放在classifier上可能會對預測產生較大的提示。

第三種attention主要想觀察用RNN所有層的hidden states計算attentiont weight，會不會比只用最後一層的還要好，結果看來是不會的。也許是因為前面幾層抓出來的feature還不夠完善，可能會成為計算attention weight時的雜訊，進而影響提取context vector的表現。

3. (30%) Beam Search

請詳細說明實做 **beam search** 的方法及參數設定，並分析結果。

從decoder的設計來看，如果將每個timestamp的output加上softmax activation，便可以得到機率$P(Y_t|Y_1...Y_{t-1})$，其中Y_i代表在timestamp i時所產出的字。

接著談談beam search的實作，步驟如下：

1. 在進入timestamp t前會有至多k條路徑$(Y_1...Y_{t-1})$被保留
2. 每條路徑會衍伸出新的k條路徑(即產生k個可能的Y_t)
3. 進入下一個timestamp前篩選這$k \times k$條路徑，僅保留最多k條路徑。篩選的公式如下：

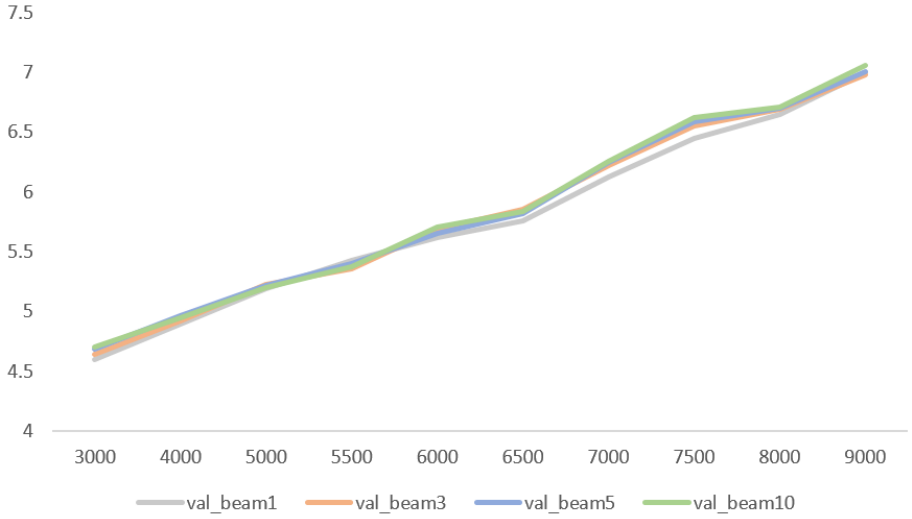
$$P(Y_1...Y_t) = P(Y_1) \times P(Y_2|Y_1) \times ... \times P(Y_t|Y_1...Y_{t-1})$$

在剪枝時前k大的路徑會被留下，但由於乘法會導致一些精度上的問題，因此改取上log後連加：

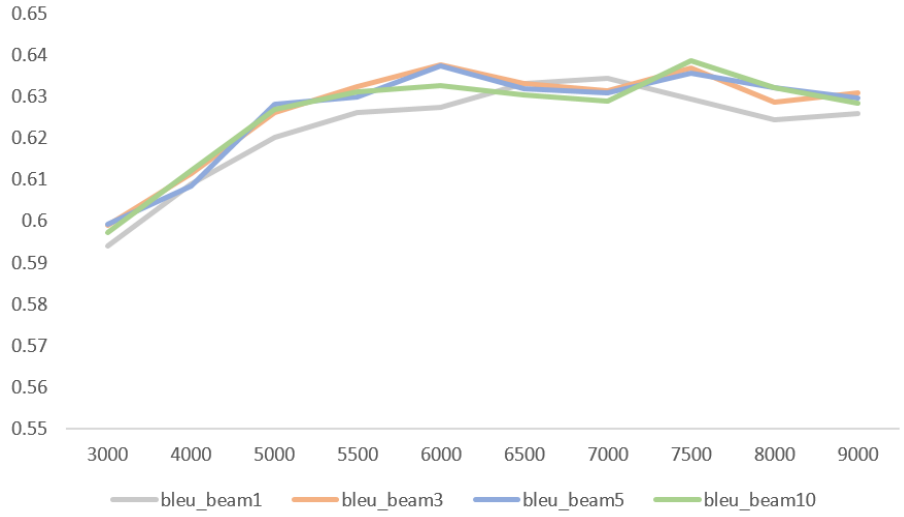
$$logP(Y_1...Y_t) = logP(Y_1) + logP(Y_2|Y_1) + ... + logP(Y_t|Y_1...Y_{t-1})$$

本題將在atten2 model上實作，主要觀察三種beam width，分別為3、5、10。以下為learning curve與BLEU score的比較：

(train與val的cross entropy loss)



(val的BLEU)



	BEAM1	BEAM3	BEAM5	BEAM10
Test BLEU	0.6073	0.6118	0.6117	0.6127

由上圖可知，beam width > 1的效果會比beam width = 1的效果還要好，但效果不是很明顯，BLEU score僅大約增加0.5%，而beam width越大也不一定保證會有越好的BLEU score，可能發生的原因如下：

1. 因為本題實作的beam search是最大化log softmax，而不是BLEU score，所以有可能提升了機率卻降低了BLEU score。
2. 某些低頻字會被轉為<UNK>，而非轉為同義詞，即使模型根據beam search丟出一個相當合理的句子，仍還是會被扣分。

```
source = ['<BOS>', 'she', 'always', 'takes', 'care', 'of', 'her', 'children', '.', '<EOS>']
pred = ['她', '總', '是', '照顧', '她', '的', '孩子', '。', '<EOS>']
target = ['她', '<UNK>', '斷', '地', '照顧', '自己', '的', '孩子', '。', '<EOS>']
```

3. BLEU score本身無法處理同義詞的部分，可能使得在算精確度時有所不足，無法完整反映模型的表現。

```
source = ['<BOS>', 'the', 'children', 'are', 'having', 'fun', 'in', 'the', 'park', '.', '<EOS>']
pred = ['孩子', '們', '在', '公園', '裏', '玩', '着', '。', '<EOS>']
target = ['孩子', '們', '在', '公園', '裏', '玩耍', '。', '<EOS>']
```

4. (20%) Schedule Sampling

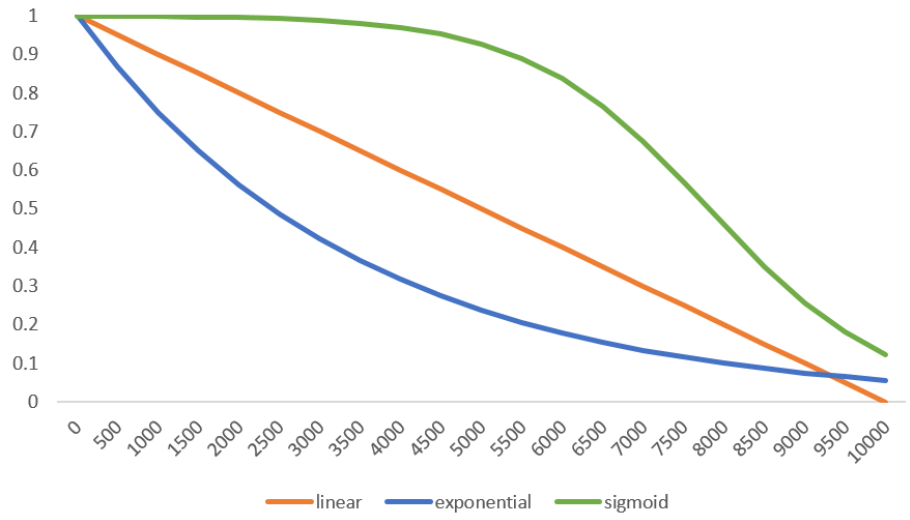
請至少實做 3 種 schedule sampling 的函數，並分析結果。

本題先用了三種schedule sampling函數，linear decay、exponential decay和inverse sigmoid decay，公式及函數圖形如下：

$$linear: \ max(0, 1 - \frac{x}{10000})$$

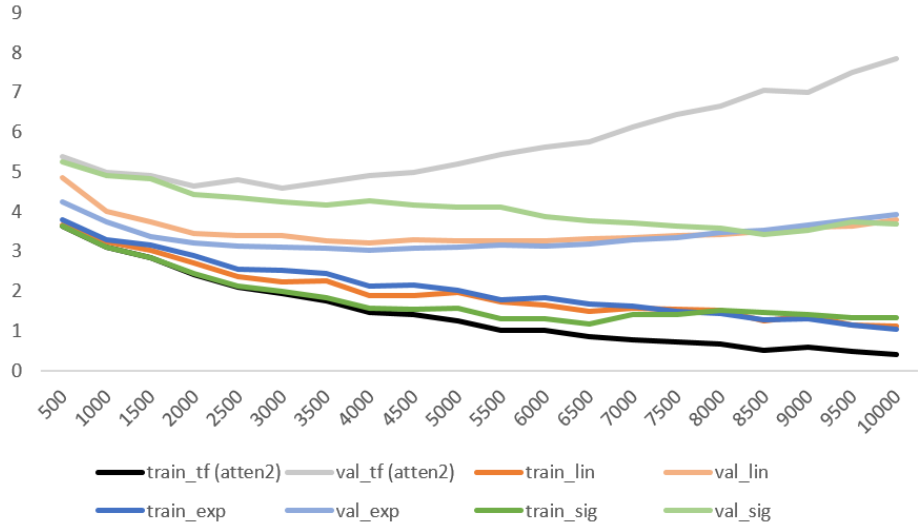
$$exponential: \ 0.75^{\frac{x}{1000}}$$

$$inverse\ sigmoid: \ \frac{1114}{1114 + exp(\frac{x}{1114})}$$

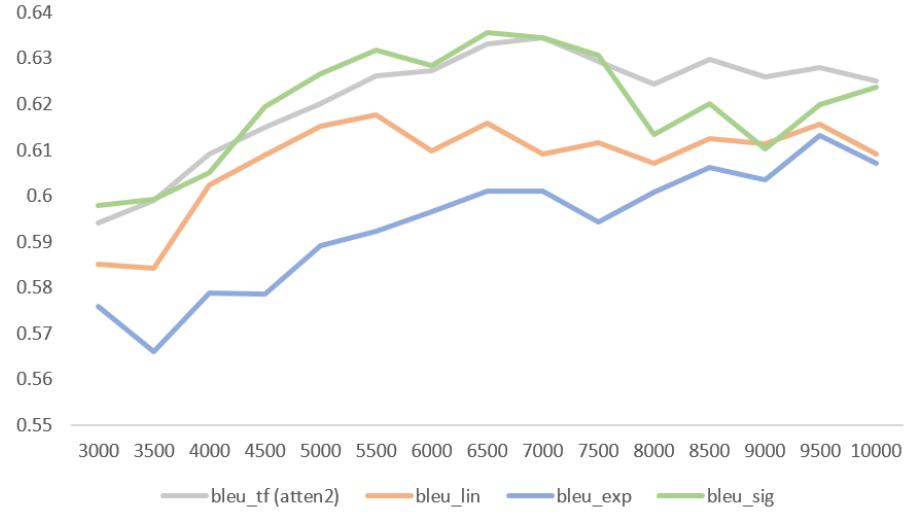


當這些scheduled sampling應用到atten2 model上，並用beam width=1進行inference時，learning curve及BLEU的比較如下(tf為teacher forcing)：

(train與val的cross entropy loss)



(val的BLEU)



	TF	LINEAR	EXP	INV SIGMOID
Test BLEU	0.6073	0.5912	0.5831	0.6042

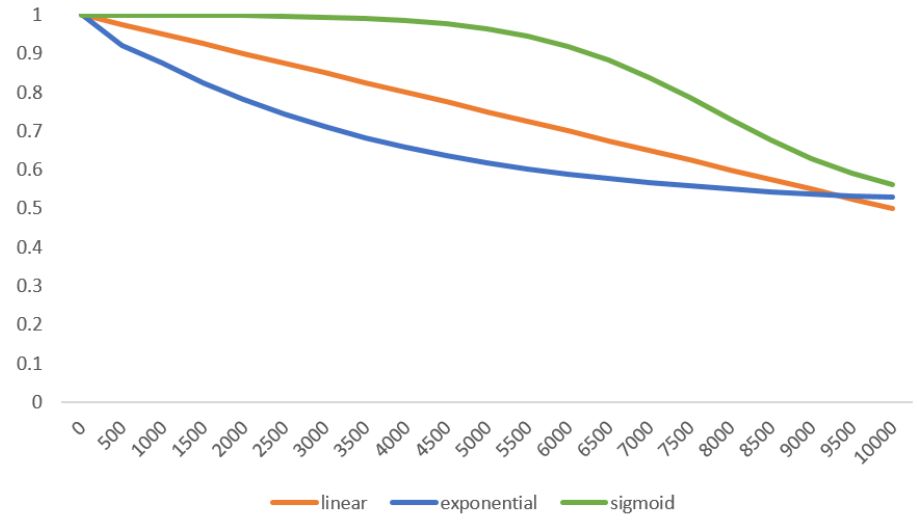
不難發現，schedule sampling的效果不如預期，training loss沒辦法像teacher forcing模型一樣低，模型的學習效率不佳，而BLEU的成績也不進則退。

就此看來，teacher forcing真的有其重要性，但由於schedule sampling確實有減輕exposure bias的道理，因此為這些函數多設了lower bound 0.5跑一些實驗，公式及函數圖形如下：

$$linear: \ max(0, 1 - \frac{x}{10000}) \times 0.5 + 0.5$$

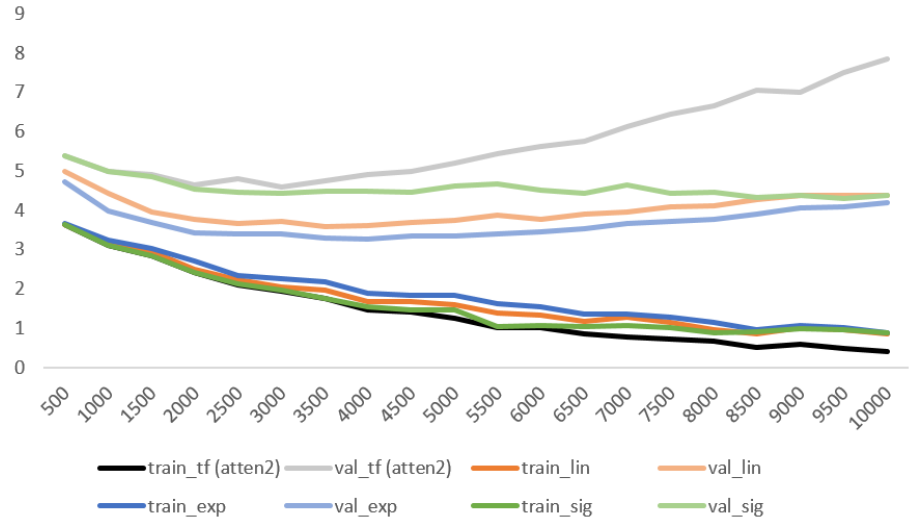
$$exponential: \ 0.75^{\frac{x}{1000}} \times 0.5 + 0.5$$

$$inverse \ sigmoid: \ \frac{1114}{1114 + exp(\frac{x}{1114})} \times 0.5 + 0.5$$

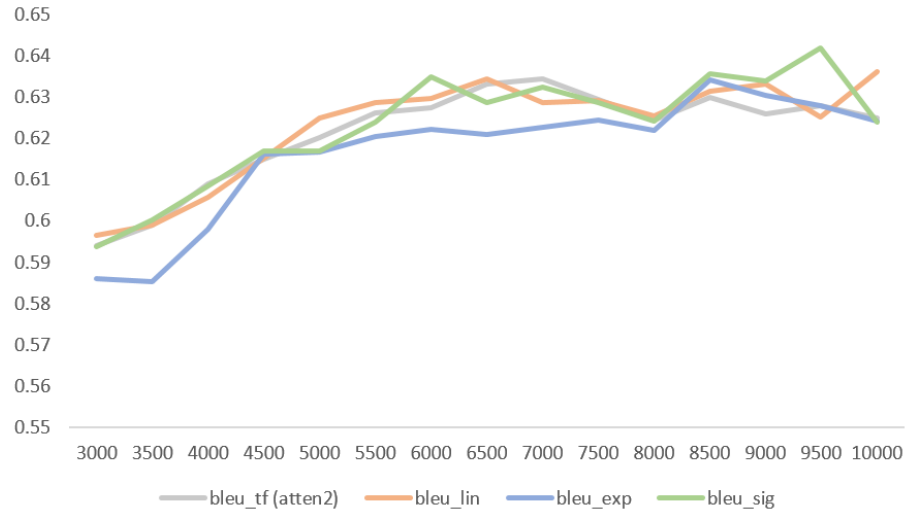


同樣將這些schedule sampling的公式應用到atten2 model(beam width=1)上，learning curve及BLEU的比較如下：

(train與val的cross entropy loss)



(val的BLEU)



	TF	LINEAR	EXP	INV SIGMOID
Test loss	7.0487	4.5970	4.1149	4.5746
Test BLEU	0.6073	0.6081	0.6038	0.6088

相比先前的schedule sampling，bounded schedule sampling的training loss有比較靠近teacher forcing model，代表teacher forcing對於模型的影響的確不容小覷，在訓練時仍需提供一定程度的teacher forcing ratio。

在testing方面，linear decay和inverse sigmoid decay的BLEU有稍微高於teacher forcing，而在loss方面則有非常顯著的進步，代表schedule sampling對模型在做inference時仍有相當正面的幫助。

最後，總結一下這三種schedule sampling函數的差異。無論有沒有使用lower bound，這些函數的表現皆為inverse sigmoid decay > linear decay > exponential decay，可能是因為模型在訓練初期預測能力還不是很好，給予較高的teacher forcing ratio可以幫助模型更快找到學習方向，收斂的效率也會更高一些。