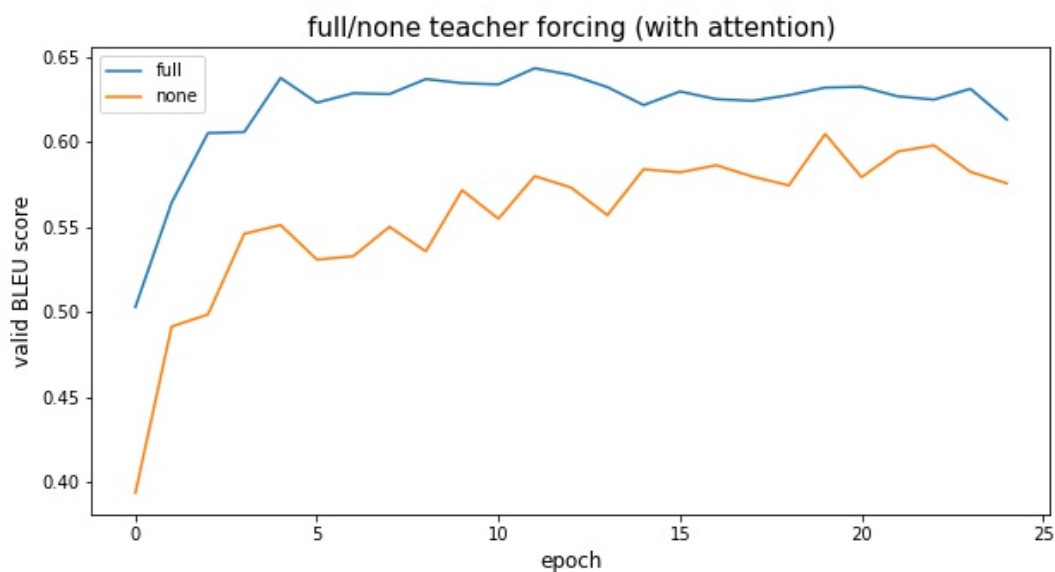


## 1. (20%) Teacher Forcing:

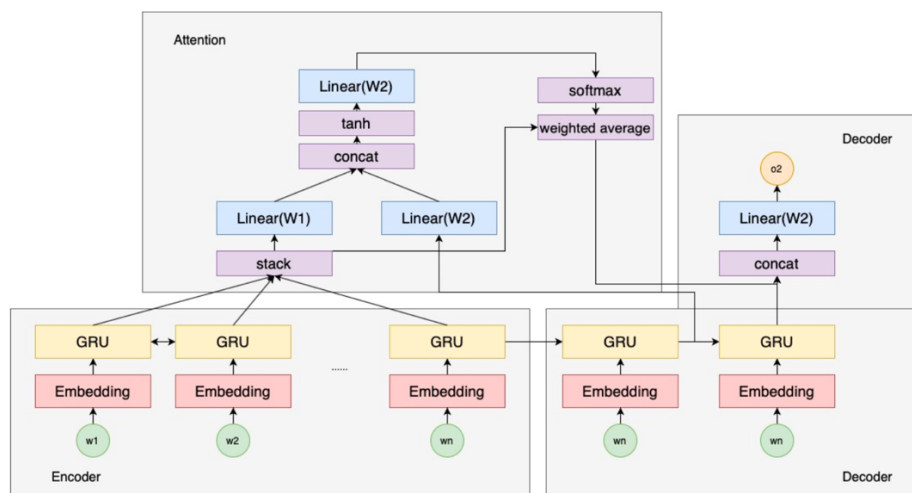
請嘗試移除 Teacher Forcing，並分析結果。



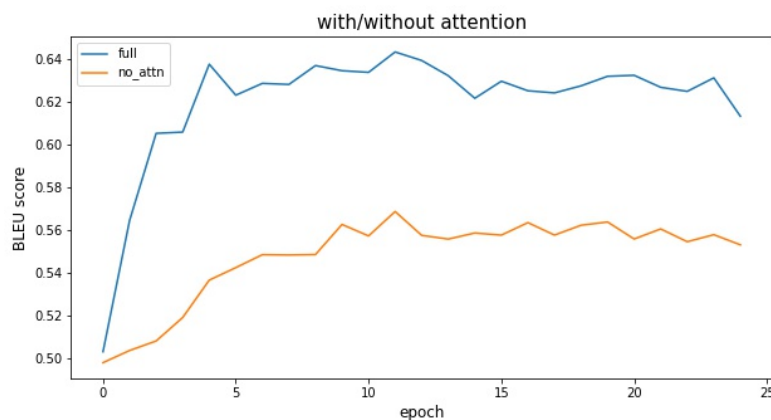
可以觀察到完全沒有 teacher forcing 的話在 valid set 和 test set 最後的結果都會比較差。

## 2. (30%) Attention Mechanism:

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

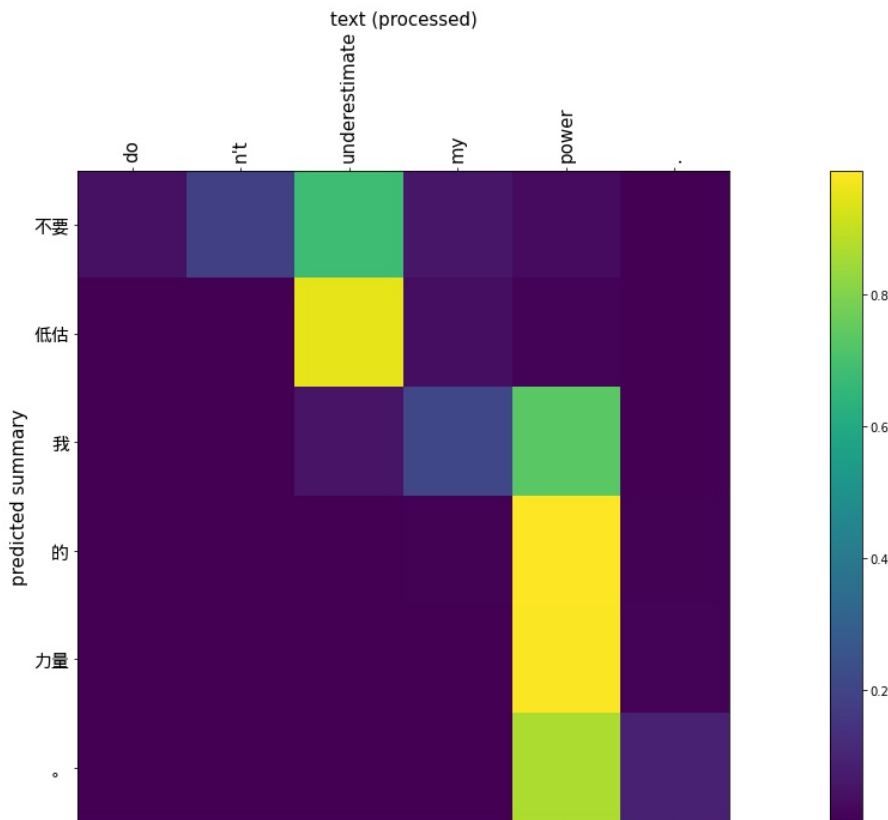


我實做的是 additive/concat attention，架構如上圖所示。訓練的結果如下：



	Full(with attn)	no_attn
Best valid BLEU	0.643	0.569
Test BLEU	0.623	0.549

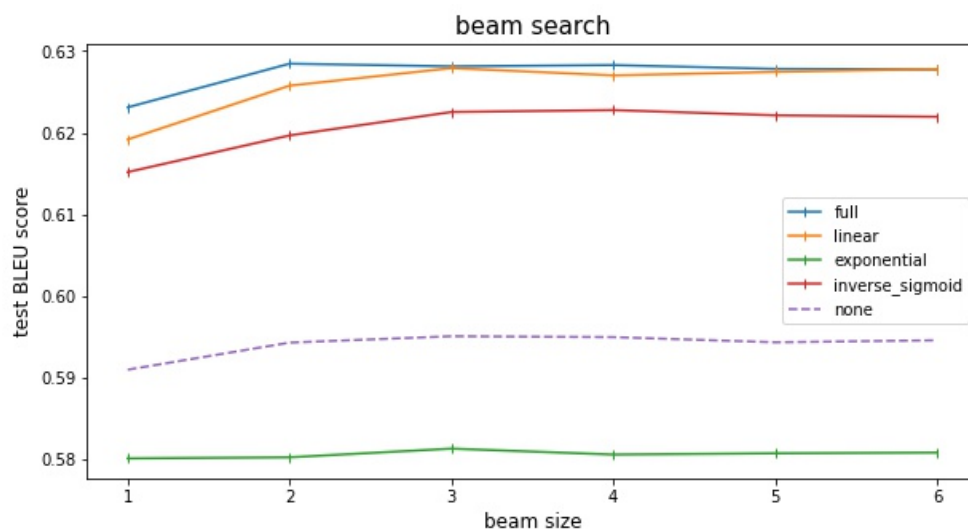
可以觀察到 attention mechanism 可以大大提升 model 的表現 (BLEU score)。



如果我們把 attention weight 畫出來的話 (如上圖)，可以發現 model 大概有抓到字的對應關係 (「低估」對到 underestimate，「力量」對到 power)。

### 3. (30%) Beam Search:

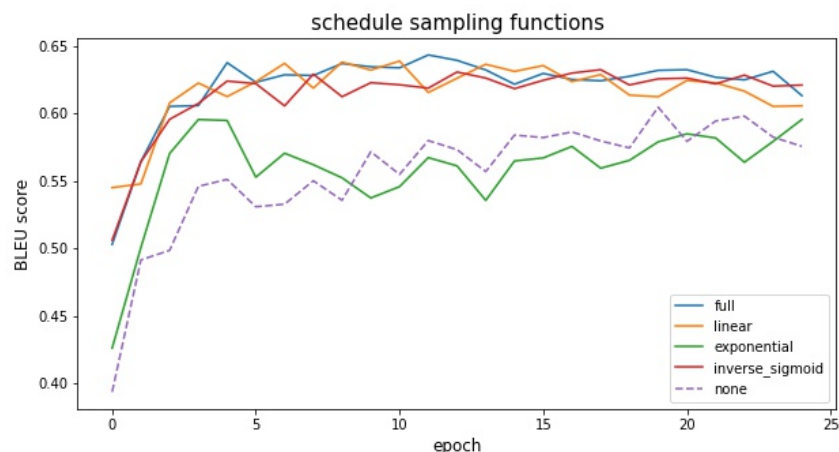
請詳細說明實做 beam search 的方法及參數設定，並分析結果。



可以觀察到對於各種不同的 model，beam search 都可以些微提升原本的 BLEU score (exponential 在圖上較不明顯，但實際上有一點點)；但隨著 beam size 的增大，marginal improvement 也逐漸變小，甚至小於零 (full 2->3, linear 3->4 等等)。這可能是因為機率最大的句子不代表他的 BLEU score 一定比較好。

#### 4. (20%) Schedule Sampling:

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



	full	linear	exponential	inv_sigmoid	none
Best valid BLEU	0.643	0.639	0.596	0.645	0.601
Test BLEU	0.623	0.619	0.580	0.615	0.591

其中 **full** 為全用 **teacher forcing**，**none** 為完全沒有 **teacher forcing**。我這邊都有使用 **attention mechanism**。

可以觀察到三種不同的 **function** 都沒有表現的比 **full** 還來得好，其中 **exp** 甚至比 **none** 還來得差一些。我想原因有一部份是這個 **dataset** 不難 **train**（連 **none** 都可以 **train** 到 0.601）。而 **exp** 這麼差的原因我猜可能是因為 **prob** 降得太快，導致 **model** 還沒學到足夠多的東西就靠自己了，這樣比從一開始就靠自己來得差（從上圖可以看到 **exp** 在大概 **epoch 5** 的時候有個陡降）。

