Q1：Data processing

使用 spacy 去做斷詞，text 的長度上限為 300 個 tokens，summary 的長度上限為 80 個 tokens，在此將得到的 tokens 以及 special_token 利用 glove.840B.300d.txt (pretrained embedding)產生相對應的 embedding matrix，而此 matrix 將在模型的 embedding layer 使用。

Q2：Extractive summarization model

(a) RNNModel ：一個 epoch 處理 batch size 篇文章

1. $\text{Embed\_}w_t$ = Dropout(Embedding Layer(w_t))， w_t 及$\text{Embed_}w_t$ 是第 t 個時間點的 token 及詞向量
2. y_t = pack_padded_sequence($\text{Embed\_}w_t$, len_text_t)，len_text_t為相對應的文章的長度
3. $output_t$, h_t, c_t= LSTM(y_t, None)，h_t及 c_t 為第 t 個時間點的 hidden 及 cell，而此 LSTM 為兩層且雙向
4. $output_t$, len_text_t = pad_packed_sequence($output_t$)
5. $pred_t$ = sigmoid(Linear(Dropout ($output_t$)))，$pred_t$為第 t 個時間點的二分類的結果。

(b) 用 validation data 去做 evaluation，根據 rouge 計算：

rouge-1：19%

rouge-2：2.76%

rouge-L：13.7%

(c) 採用 BCEWithLogitsLoss 當作 Loss Function

Loss (pred, target) = mean{$l_0, l_1, l_{2,\ldots} l_{n-1}$}

$l_m = -(pred_m * \log(\delta(target_m)) + (1 - pred_m) * \log(1 - \delta(target_m)))$，n 為資料數量，$pred_m$為第 m 筆資料的預測，$taregt_m$為第 m 筆資料的標籤，$\delta$為 sigmoid。

(d) 採用 Adam 當作 optimization alogorithm，learning rate 為 0.001 且 batch size 為 128。

(e) pred's shape = [batch size, the number of the tokens of the text]，各文章的 token 包含數量最多的句子即為 summary。

Q3：Seq2seq + Attention model

(a)

- Encoder：一個 epoch 處理 batch size 篇文章

1. $\text{Embed\_}w_t$ = Embedding Layer(w_t)， w_t 及$\text{Embed_}w_t$ 是第 t 個時間點的 token 及詞向量
2. y_t = pack_padded_sequence($\text{Embed\_}w_t$, len_text_t)，len_text_t為相對應文章的長度

3. $output_t$, h_t= GRU(y_t, None)，$output_t$為第 t 個時間點的 context vector，而此 GRU 為一層且雙向
4. $output_t$, len_text_t = pad_packed_sequence($output_t$)
5. $final\_h_t$ = Tanh(Linear ($hidden_t$))，$hidden_t$為第 t 個時間點最後的 forwards 和 backwards 的 hidden 所組成，$final_h_t$為 deocder 的 input hidden。

- Attention：

$attention_t$ = softmax(Linear(h_t 和 c_t 接在一起))，h_t為 encoder 的輸出 hidden，也就是上述的$final_h_t$，c_t為 encoder 的 context vector，也就是上述的$output_t$，其中利用 mask_fill 在 mask 為 1 時用 value 做填充

- Decoder：對於一篇文章的摘要生成是一次一個 token

1. Embed_$target_t$ = Embedding Layer($target_t$)，$target_t$ 及Embed_$target_t$ 是第 t 個時間點的 target 及其詞向量
2. $weight_t = attention_t * output_t$ (Batch Matrix Multiply, bmm)，$output_t$為 encoder 的輸出，$attention_t$為相對$output_t$應該放多少的注意力，$attention_t$的數值介於 0 到 1
3. $decoder\_input_t =$ Embed_$target_t$ 和$weight_t$ 接在一起
4. $pred_t$, h_t= GRU($decoder\_input_t$, h_{t-1})，h_t為第 t 個時間點的 hidden，$pred_t$經過 linear 處理後會存入 prediction，而此 GRU 為一層且單向
5. prediction = [length of target, batch size, vocab size]，經由 softmax 並挑選第三維機率最大的 token 組成 summary

(b) 用 validation data 去做 evaluation，根據 rouge 計算：

rouge-1：22.5%

rouge-2：5.9%

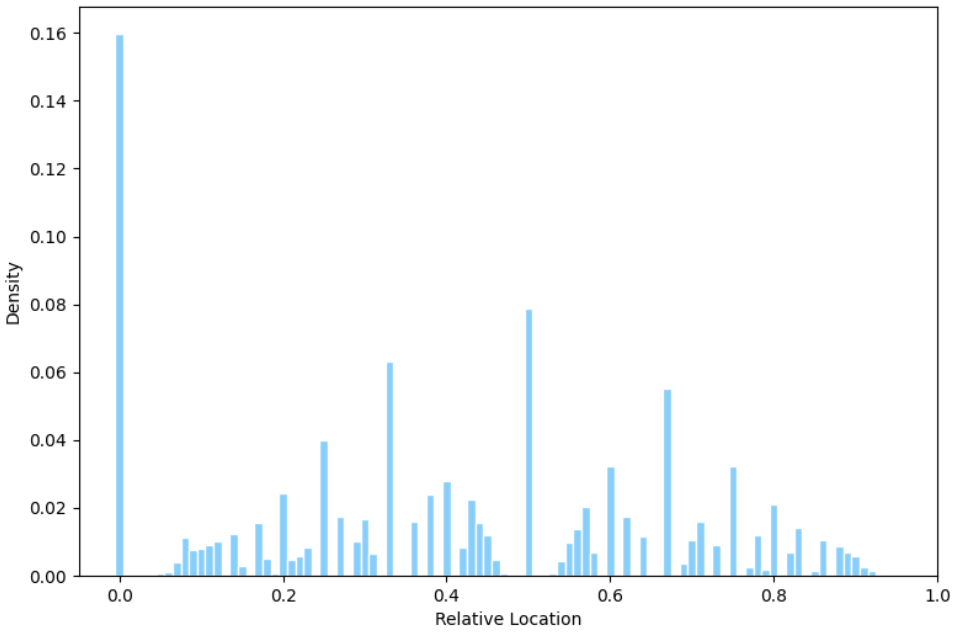
rouge-L：19.4%

(c) 採用 CrossEntropyLoss 當作 loss function

Loss (pred, target) = $-\log(\frac{\exp(pred[target])}{\sum_j \exp(pred[j])})$

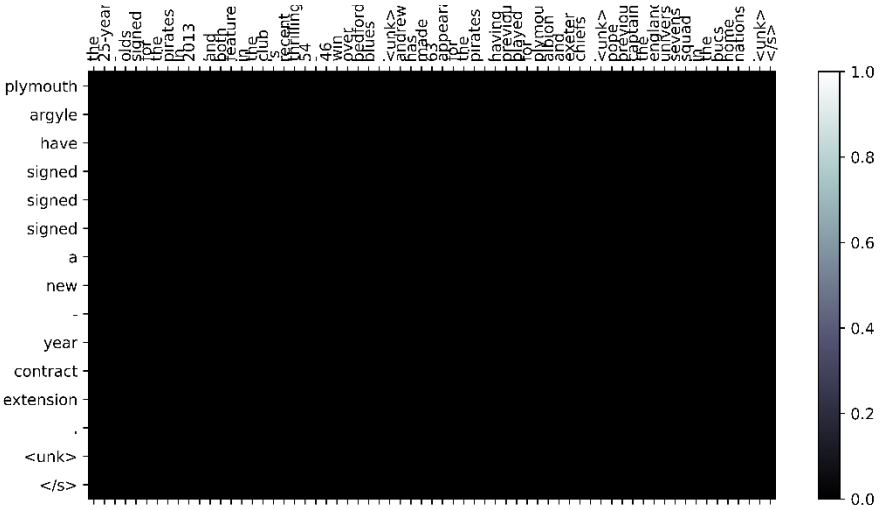
(d) 採用 Adam 當作 optimization alogorithm，learning rate 為 0.001 且 batch size 為 32。

Q4：The distribution of relative locations



根據上面這張圖可以發現相對位置為 0 的密度最高，這也符合了一般人對於英文文章的認知，topic sentence 通常都是文章的第一句，適合做 summary。

Q5：Visualize the attention weights



由於 attention + seq2seq 的模型訓練上沒有達到預期效果，所以畫出的 attention weight 呈現 0 的狀態，但在這裡要多加說明，此圖的 x 軸為 input tokens，y 軸為 output tokens，圖中的顏色代表 weight，顏色越白代表預測此字時會越專注於輸入的 input token。

Q6：Rouge-L

Rouge-L 使用了最長共同子序列(LCS)。他的計算方式為下列：

$$R_{lcs} = \frac{LCS(X,Y)}{m} \quad (1)$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \quad (2)$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \quad (3)$$

X = $[x_1, x_{2,\dots}x_m]$，是參考摘要(Reference Summaries)的子序列

Y = $[y_1, y_{2,\dots}y_n]$，是自動摘要 (Predict Summaries)的子序列

m 為參考摘要的長度

n 為自動摘要的長度

數學式(1)可以說是 Recall，數學式(2)可以說是 Precision，數學式(3)就是所謂的 Rouge-L，而式子中的 β 在 DUC 中通常都是一個很大的常數，所以 Rouge-L 通常都只考慮R_{lcs}。