2023 年臺灣國際科學展覽會
優勝作品專輯

作品編號　190012

參展科別　電腦科學與資訊工程

作品名稱　應用深度學習 sequence to sequence model 於古文解譯

得獎獎項　四等獎

就讀學校　臺北市立第一女子高級中學

指導教師　陳縕儂、張清俊

作者姓名　楊予瑄、劉又甄

關鍵詞　深度學習、古文、翻譯

作者簡介

大家好，我們是來自北一女中科學班的劉又甄和楊予瑄。我們自從高中接觸了程式語言之後，就非常有興趣。閱讀與理解古文是學習過程中不可或缺的課程，但古文的難以理解成為許多人學習上的阻礙。因此，我們希望能建立一套完善的古文解譯系統，減少大眾與古文的距離感，增進學習古文的意願。感謝陳縕儂教授一路上的悉心指導，雖然過程中充滿過程與挑戰，但從錯誤中學到的經驗更加珍貴。

摘要

以將古文翻譯成白話文為初衷，以爬蟲擷取古文解譯網站「讀古詩詞網」中的大量古文及其白話翻譯作為訓練用的資料，並按照不同文體分開訓練。我們先嘗試用 Bert 模型做選擇題：給一句古文讓機器從四個選項中選出其翻譯。一開始隨機挑選其餘三個選項，正確率高達 96%。因此我們挑戰更困難的設置，撰寫搜尋關鍵字的程式，將有與題目古文相同字的白話文放入選項。雖然準確率有些許降低，但仍高於只選重複字最多選項的結果，代表模型有發展出獨立的判定標準。選擇題成功後，我們用 MT5 模型嘗試更困難的翻譯，並在訓練集中新增提供不同前後文的注釋資料幫助訓練。雖然還無法翻得非常準確，但仍在某些句子有不錯的表現。我們也發現了模型對某些特定類型字詞的翻譯有待加強，未來希望透過加強代名詞判斷訓練及持續新增注釋來增加整體翻譯能力。

Abstract

With the intention of translating ancient texts into vernacular texts, a crawler program was written to capture ancient texts and their vernacular translations from the ancient text translation website "Du Gu Shi Cih Website ", so as to obtain a large number of training materials, and train them separately according to different literary styles. We first tried to do multiple choice questions using a Bert (Bidirectional Encoder Representations from Transformers) model: give an ancient text, and let the machine choose its translation from four options. The other three options were random at the beginning, and the accuracy rate was as high as 96%. Our method achieved good results in random settings. Therefore, we challenged the more difficult experimental settings. By writing another program to search keywords, we found out specific vernacular texts with the same words as each line of the ancient texts and put them into the options according to the number of repeated words. Although the accuracy rate was slightly reduced, the final accuracy rate was still higher than that of only selecting options including the most repeated words, which means the model didn't just pick the option with the most repeated words, but had developed its own independent judgment criteria. After the multiple-choice question was successful, we tried more difficult translations. After trying a variety of models, the model with the best results was MT5 (A massively multilingual pre-trained text-to-text transformer). Moreover, we added annotation data accompanied by different contexts to the training set to help training. Although we haven't succeeded in translating the ancient texts very accurately, our model still performed well in some sentences. We also found that the model's translation of certain types of words needs to be strengthened. In the future, we hope to increase the overall translation ability by strengthening pronoun judgment training and continuously adding annotations.

壹、研究動機

閱讀與理解古文是學習過程中不可或缺的課程，然而因時代差異，古文的詞彙表達常與現代用語有相當大的差異，對古文的難以理解便成為許多人學習上的阻礙。因此，我們希望能建立一套完善的古文解譯系統，並幫助我們透過熟悉的語言理解其文意，減少大眾與古文的距離感，增進學習古文的意願。目前在市面上已經有各種不同語言間的翻譯軟體，但單一語言的古今用語差異通常文字密度不一，這類的古文解譯則很少人研究過；而且世界各地的語言都會有古今用語差異，若我們能建立出完善的古文解譯系統，未來也有機會應用於其他語言上。

貳、研究目的

綜合上述討論，本研究的目的為「利用深度學習創建一個將輸入的古文翻譯成白話文的完善系統」，可細分為下列項目：

一、研究機械學習模型(Transformer)的運作機制及特性

二、 研究不同種類模型的差異，並試出最適合用於古文翻譯的模型

三、 調整模型內的不同參數以獲得最準確且通順的翻譯結果

四、 將最佳翻譯之模型設計成網頁或應用程式，供需要的人使用

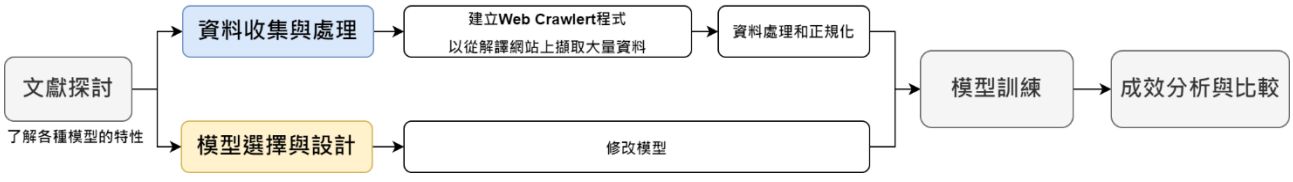
參、研究設備及器材

一、硬體:筆記型電腦

二、軟體及工具:Google Colaboratory

肆、研究過程或方法

一、研究流程與架構



圖一、研究流程圖

二、研究文獻探討

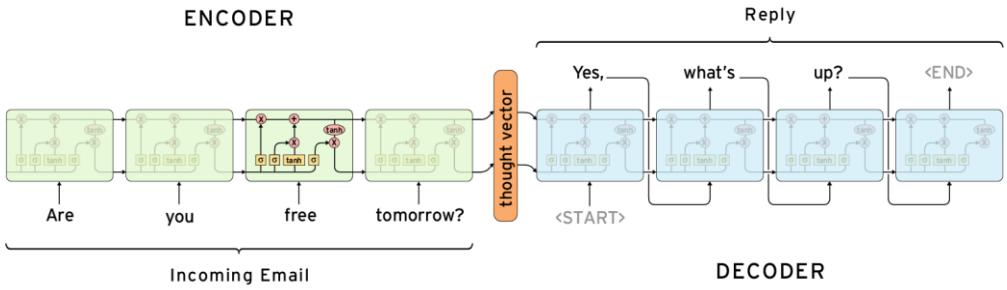
(一)Seq2seq (Sequence-to-Sequence)

顧名思義 ，Seq2seq 是一種將一個序列轉換為另一序列的建模方法，輸入和輸出是兩個向量序列，經常使用 Encoder-Decoder 的架構，使用 Encoder 來對輸入序列編碼並將編碼結果輸入至 Decoder 進行解碼。常見於機器翻譯、自然語言生成等任務。

1. 模型架構

原始的 Seq2seq 是由 Encoder 與 Decoder 兩個 RNN 構成。此外也有使用到 Self-Attention Layer 的 Seq2seq，也就是 Transformer 的基本架構。

Encoder 負責將輸入序列轉換成一個向量，這個向量會囊括原序列的重要訊息，我們通常把這個向量稱為 context vector；相反地， Decoder 則是根據 context vector 來生成文字。



圖二、Sequence-to-Sequence 模型架構示意圖

(圖片來源： http://zake7749.github.io/2017/09/28/Sequence-to-Sequence-tutorial/)

2. 訓練方式

Seq2seq 模型最常見的訓練方式為 teacher-forcing，此訓練方式將輸入和輸出序列分別輸入給 Encoder 和 Decoder，並要求 Decoder 直接輸出目標輸出序列，訓練的目標為最大化目標序列的機率，通常使用 cross-entropy 作為訓練的損失函數。

對於輸入序列 $x_1,\cdots,x_T$與輸出序列 $y_1,\cdots,y_{T'}$ 而言，透過 Encoder 我們能將 $x_1,\cdots,x_T$ 轉換成 context vector v，我們希望能在 Decode 階段最大化條件機率 p：

$$p(y_1,\ldots,y_{T'}|x_1,\ldots,x_T)=\prod_{t=1}^{T'}p(y_t|v,y_1,\ldots,y_{t-1})$$

(公式來源： http://zake7749.github.io/2017/09/28/Sequence-to-Sequence-tutorial/)

最外圍的連乘符號表示想求的是全局最優解，而內部的機率項則指的是對時間點 t 而言，模型知道已經聽了什麼(v, context vector)，以及之前說了些什麼($y_1,\cdots,y_{t-1}$)，並以這兩件事為基準，來評估現在該說什麼 (y_t)。

3. 自迴歸模型(Autoregressive model)

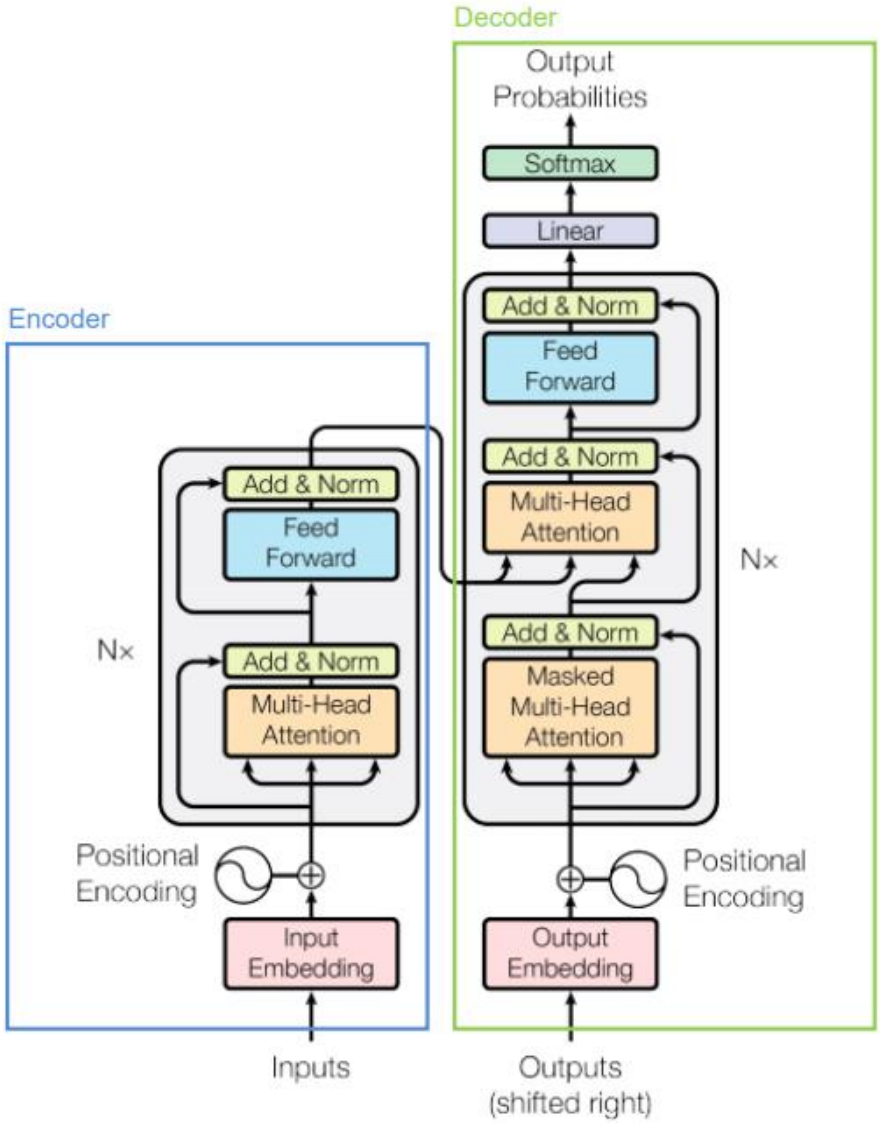
Seq2seq 模型通常是一種自迴歸模型，由於在實際測試時，我們僅知道輸入序列而不會先知道輸出序列為何，因此無法像訓練時一樣將輸出序列直接同時輸入給 Decoder，此時通常會使用自迴歸解碼，首先將一個 Start token 輸入給 Decoder，

讓 Decoder 輸出第一個字，再將這個字作為 Decoder 的第二個輸入，得到輸出的第二個字，重複這個動作直到 Decoder 輸出結束。

(二)Transformer

Transformer 是基於 Self-attention 架構的 Seq2seq 模型 ，其中 Attention 架構結合了 CNN 及 RNN 的優點，並且能改善 RNN 不能平行運算的缺點。

Transformer 由 Encoder 和 Decoder 組成，架構如圖五所示。Encoder 是下圖左方的部分，Decoder 是下圖右方的部分。



圖三、Transformer 模型架構示意圖
(圖片來源：Attention Is All You Need)

1. Encoder

輸入x_i 加上 position embedding 後，依序經過以下步驟：

(1) Multi-head attention

(2) Add & Norm

Add 即是把 (1)得到的 Attention 的輸出各自加上原本對應的輸入，後面的 Add 也都是和這裡相同：把輸入和輸入加起來。

Norm 指的是 Layer Normalization：將向量內的元素變成 mean = 0，variance = 1。

(3) Feed forward layer、Add & Norm

以上的程式塊有 N 層，一層的輸出會成為下一層的輸入，最後的輸入再傳到 Decoder。

2. Decoder

Decoder 的輸入是上一個時間戳記的輸出。一樣加上 position embedding 後，依序經過：

(1) Masked multi-head attention

Masked 的意思是只會關注在已經產生的序列。

(2) Add & Norm

(3) Multi-head attention、Add & Norm

這裡的 Attention 會關注到 encoder 的輸出，也就是把(2) 的輸出拿去算 query vector。key vector 和 value vector 則用 encoder 的輸出來計算。

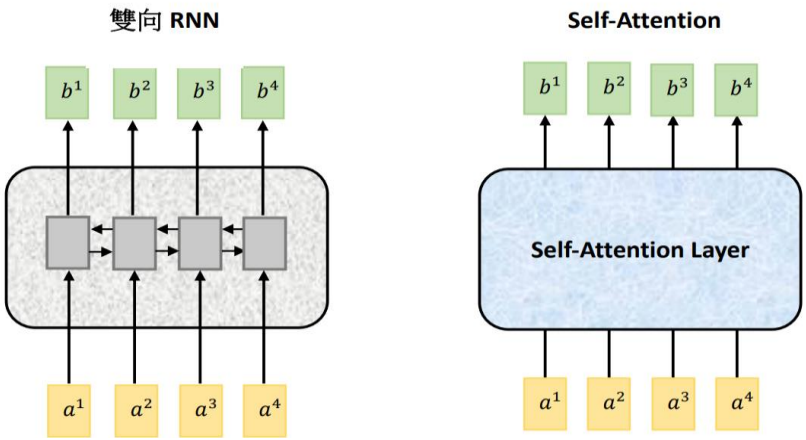
(4) Feed forward layer、Add & Norm

以上 程式塊一樣也有 N 層，最後的輸出再根據任務進行不同的操作。

3. Self-Attention 自注意力機制

(1) 模型架構

Self-Attention 是一個能用來取代 RNN 的機制。其架構如圖所示：



圖四、雙向 RNN 及 Self-Attention 架構

(圖片來源： https://www.youtube.com/watch?v=ugWDIIOHtPA)

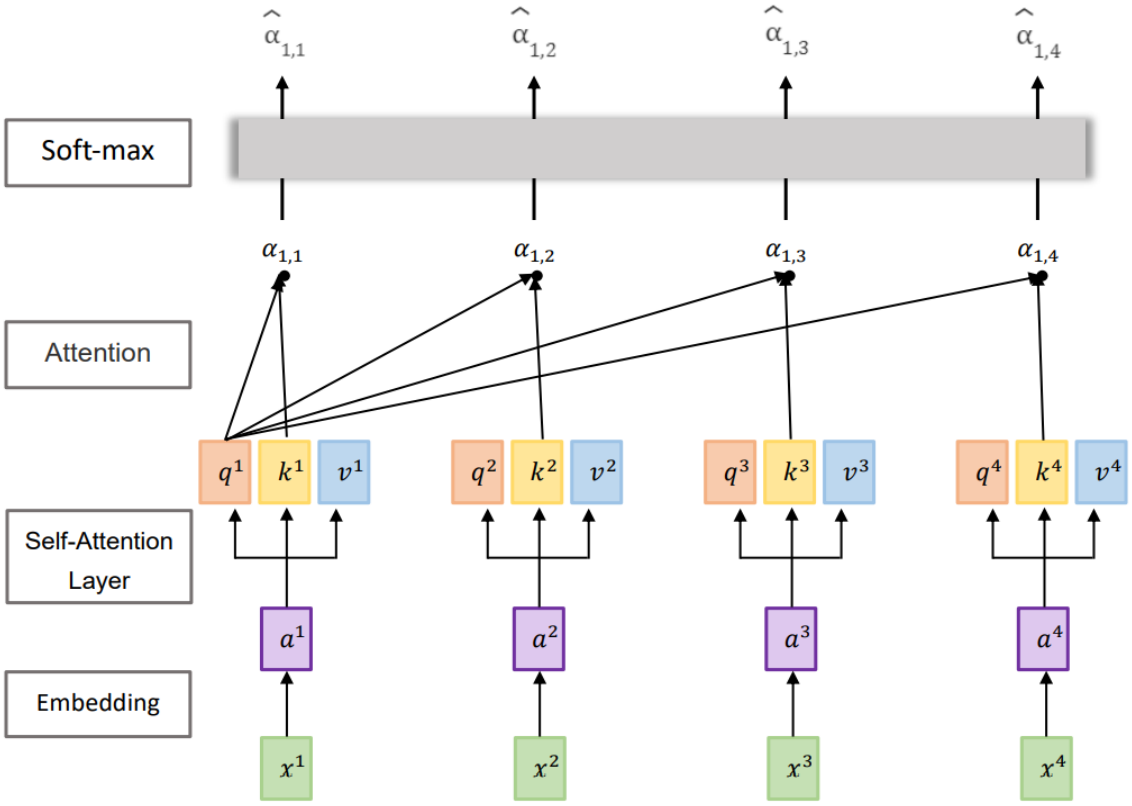
輸入一個向量序列，經過 Self-Attention Layer 的計算，產生對應的向量序列作為輸出。

假設輸入的中的元素依序為a^1、a^2、a^3、a^4，輸出的向量序列中的元素依序為b^1、b^2、b^3、b^4。圖左的輸出為經過雙向 RNN 計算後的結果，圖右的輸出則為經過 Self-Attention Layer 計算後的結果。

與雙向 RNN 相同的是，經過 Self-Attention Layer 計算後輸出的序列中的每一個元素(b^1、b^2、b^3、b^4)都是考慮過輸入向量序列中所有元素(a^1、a^2、a^3、a^4)的結果，故 Self-Attention 與雙向 RNN 可達成相同的建模效果。

但與雙向 RNN 不同的是，Self-Attention 輸出的b^1、b^2、b^3、b^4都是同時被計算出來的；而用 RNN 輸出時，會依序計算出b^1、b^2、b^3、b^4，也就是說，輸出序列中的每一個元素都必須在上一個元素被計算出來之後才能被計算輸出，會耗費大量時間。利用 Self-Attention 則可利用現代 GPU 的平行計算能力大量縮減計算時間，故其常用於取代 RNN。

(2) Self-Attention 的計算處理



圖五、Self-Attention 流程圖

(圖片來源： https://www.youtube.com/watch?v=ugWDIIOHtPA)

假設輸入序列中的元素為x^1、x^2、x^3、x^4，處理步驟如下：

a. Embedding

將x^1、x^2、x^3、x^4乘上一個權重矩陣 W，轉換為對應的詞向量 a^1、a^2、a^3、a^4。

其公式為：$a^i = Wx^i$

b. Self-Attention Layer

將a^1、a^2、a^3、a^4作為 self-attention layer 的輸入，分別經過三個不同矩陣的轉換：W^q、W^k、W^v，產生三個不同的：q 向量、k、v。其中，q 代表 query，k 代表 key，v 代表 value。

其公式為：

$q^i = W^q a^i$ ， $k^i = W^k a^i$ ， $v^i = W^v a^i$

c. Attention

將每個 q 對所有 k 計算注意力的權重。其公式為：

$\alpha_{1,i} = q^1 \cdot k^i / \sqrt{d}$ ，其中 d 代表 q 和 k 的維度。

d. Softmax

將輸入的向量轉為機率的形式輸出，其中輸出的各元素值在 0 到 1 間，各元素值之總和為 1。這邊使用 softmax 的目的為 attention weight normalization，我們可以將經過正規化的權重視為每個 key 對於此 query 的重要程度。

接著計算v、$\hat{a}$的 weighted sum，就得到b，b集合成的矩陣就是輸出的矩陣。

其公式為：

$$\hat{\alpha}_{1,i} = \exp(\alpha_{1,i}) / \sum_j \exp(\alpha_{1,j})$$

$$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$$

(三)Bert : Self-supervised Learning (自監督學習)的應用

Bert 模型開啟了將神經網路使用大量未標註資料進行 Pretraining (預訓練)，再將預訓練後的參數使用在目標任務上進行 Fine-tuning (微調) 的熱潮。

1. 預訓練 (Pretraining)

Bert 在 pre-train(預訓練)的時候，訓練了兩個自監督的任務：

Masked Language Modeling 跟 Next Sentence Prediction

(1) Masked Language Modeling:

把一句句子中的幾個字隨機遮住 (變成特殊的[MASK]token)，然後預測這些被遮住的字是哪些字。在 BERT 中，每個 token 有 15% 的機率被代換，被代換的 token 有 80% 的機率變成[MASK]，有 10% 的機率變成隨機 token，有 10% 的機率不變。讓模型去預測原本是什麼 token，用 cross entropy 訓練。

(2) NEXT SENTENCE PREDICTION:

給兩個句子，預測第二句是不是第一句的下一句。是跟不是各抽取 50%。

2. 模型架構

Bert 是由經過訓練的 Transformer 的 Encoder 堆疊而來，有兩種 size 分別是: base 和 large，兩種 BERT 模型大小都有大量的 Transformer Blocks(self-attention layers) Base 版本有 12 個，Large 版本有 24 個。

一般的 Bert 模型分成兩部分:

(1) Transformer Encoder:

處理輸入的句子，並將它從句子中提取的一些信息傳遞給下一個模型。

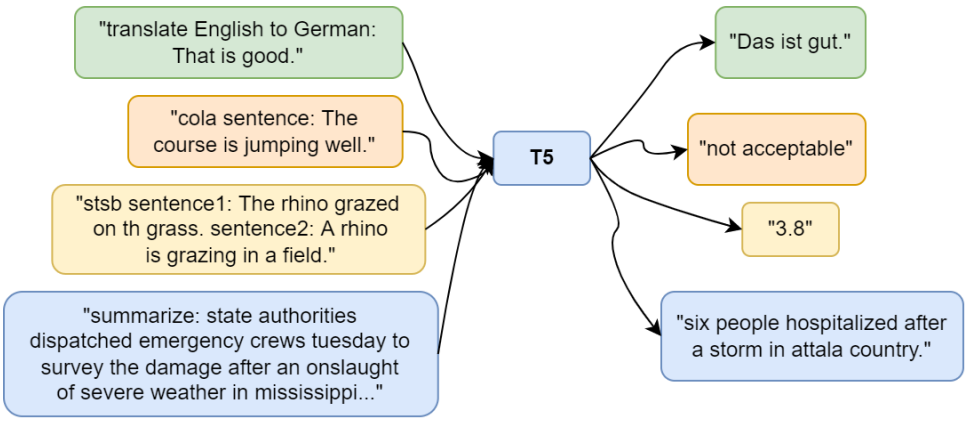
(2) Logistic Regression:

處理輸出結果並且將句子進行分類，輸出 0 或 1。

(四)T5 (TEXT-TO-TEXT TRANSFER TRANSFORMER)

1. 模型簡介

T5 模型最大的特點就是把所有的自然語言的任務，凡舉翻譯、問答、情感偵測

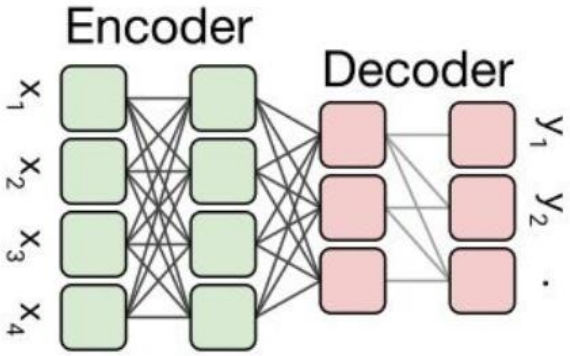


圖六、T5 示意圖

(圖片來源： https://codingnote.cc/zh-tw/p/19594/)

總結等任務，都以 sequence to sequence 的方式解決，如圖所示：

2. 模型架構



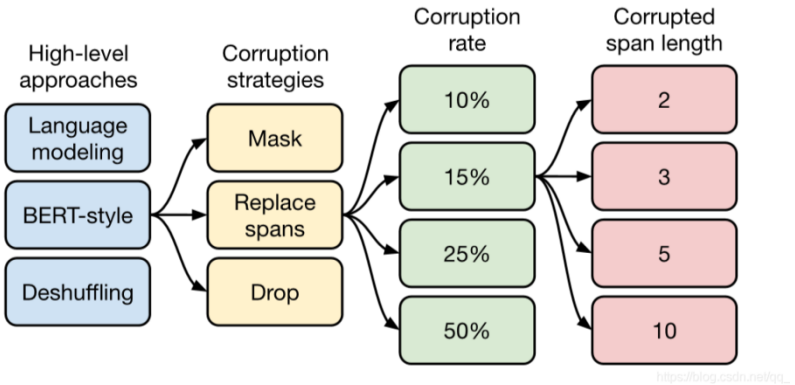
圖七、T5 架構

(圖片來源： https://codingnote.cc/zh-tw/p/19594/)

如圖九，分成 Encoder 和 Decoder 兩部分，對於 Encoder 部分，輸入可以看到全體，之後結果輸給 Decoder，而 Decoder 因為輸出方式只能看到之前的。

3. 預訓練(Pretraining)

總共從四方面來進行比較，如圖所示：



圖八、T5 預訓練

(圖片來源： https://codingnote.cc/zh-tw/p/19594/)

(1) 第一個方面，高層次方法（自監督的預訓練方法）對比，總共三種方式：

a. 語言模型式，使用各種統計和概率技術來確定給定單詞序列在句子中出現的概率。

b. BERT-style 式，就是像 BERT 一樣將一部分破壞掉，然後還原出來。

c. Deshuffling （順序還原）式，就是將文本打亂，然後還原出來。

最後發現 BERT-style 式 結果最好。

(2) 第二方面，對文本一部分進行破壞時的策略，也分三種方法：

a. Mask 法，如現在大多模型的做法，將被破壞 token 換成特殊符如[M]；

b. replace span（小段替換）法，可以把它當作是把上面 Mask 法中相鄰[M]都合成了一個特殊符，每一小段替換一個特殊符，提高計算效率；

c. Drop 法，沒有替換操作，直接隨機丟棄一些字符。

最後發現 replace span 結果最好。

(3) 第三方面，對文本百分之多少進行破壞，挑了 4 個值，10%、15%、25%、50%，最後發現 15% 結果最好。

(4) 第四方面，因為 Replace Span 需要決定對大概多長的小段進行破壞，挑了 4 個值，2、3、5、10 這四個值，最後發現 3 結果最好。

4. 完整 T5 模型

(1) Transformer Encoder-Decoder 模型

(2) BERT-style 式的破壞方法

(3) Replace Span 的破壞策略

(4) 15 %的破壞比

(5) 3 的破壞時小段長度

三、研究方法設計

(一)資料收集與整理

1. 資料收集

因訓練需要大量資料，以人工方式收集缺乏效率及可行性，所以我們製作了 Web Crawler 的程式，從一集結了大量人工古文解譯資料的網站「讀古詩詞網」(https://fanti.dugushici.com/)上，擷取大量訓練用資料。

網站介面如下(以文言文「賣油翁」為例)：

詩詞名	朝代	作者	正文
伶官傳序	宋代	歐陽修	嗚呼！盛衰之理，雖曰天命，豈非人事哉！原莊宗之所以得天下，與其所以失之者，可以知之矣。 世言晉王之將終也...
莊暴見孟子	先秦	孟子及弟子	莊暴見孟子，曰：“暴見於王，王語暴以好樂，暴未有以對也，”曰：“好樂何如？” 孟子曰：“王之好樂甚，則齊...
突秋	先秦	孟子及弟子	孟子曰：“無或乎王之不智也，雖有天下易生之物也，一日暴之，十日寒之，未有能生者也。吾見亦罕矣，吾退而寒之者至矣，吾...
齊人有一妻一妾	先秦	孟子及弟子	齊人有一妻一妾而處室者，其良人出，則必饜酒肉而後反，其妻問所與飲食者，則盡富貴也，其妻告其妾曰：“良人出，則必饜...
齊桓晉文之事	先秦	孟子及弟子	齊宣王問曰：“齊桓、晉文之事，可得聞乎？” 孟子對曰：“仲尼之徒，無道桓、文之事者，是以後世無傳焉，臣未...
孟子見梁襄王	先秦	孟子及弟子	孟子見梁襄王，出，語人曰：“望之不似人君，就之而不見所畏焉。 卒然問曰：‘天下惡乎定？’ 吾對...
柳毅傳	唐代	李朝威	儀鳳中，有儒生柳毅者，應舉下第，將遽湘濱，念鄉人有客於涇陽者，遂往告別。至六七裏，鳥起馬驚，疾逸道左，又六七裏，乃...
項脊軒志	明代	歸有光	項脊軒，舊南閣子也。室僅方丈，可容一人居，百年老屋，塵泥滲漉，雨澤下注；每移案，顧視，無可置者。又北向，不能得日，...
爲學	清代	彭端淑	天下事有難易乎？爲之，則難者亦易矣；不爲，則易者亦難矣。人之爲學有難易乎？學之，則難者亦易矣；不學，則易者亦難矣，...
賣油翁	宋代	歐陽修	陳康肅公堯咨善射，當世無雙，公亦以此自矜。嘗射於家圃，有賣油翁釋擔而立，睨之，久而不去，見其發矢十中八九，但微頷...

上一頁 1 2 ... 10 11 12 13 14 15 16 17 18 ... 42 43 下一頁　共426首

康肅公陳堯諮善於射箭，世上沒有第二個人能跟他相媲美，他也就憑着這種本領而自誇。曾經（有一次），（他）在家裏（射箭的）場地射箭，有個賣油的老翁放下擔子，站在那裏斜着眼睛看着他，很久都沒有離開。賣油的老頭看他射十箭中了八九成，但只是微微點點頭。

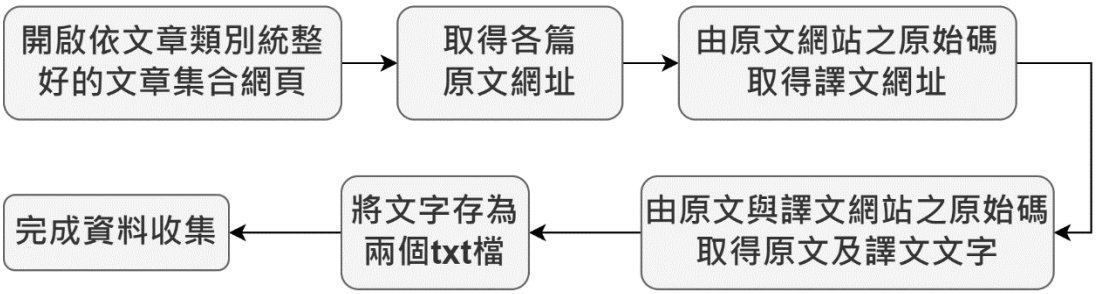
陳堯諮問賣油翁：”你也懂得射箭嗎？我的箭法不是很高明嗎？”賣油的老翁說：”沒有別的（奧妙），不過是手法熟練罷了。”陳堯諮（聽後）氣憤地說：”你怎麼敢輕視我射箭（的本領）！”老翁說：”憑我倒油的經驗就可以懂得這個道理。”於是拿出一個葫蘆放在地上，把一枚銅錢蓋在葫蘆口上，慢慢地用油杓舀油注入葫蘆裏，油從錢孔注入而錢卻沒有溼。於是說：”我也沒有別的（奧妙），只不過是手熟練罷了。”陳堯諮笑着將他送走了。

這與莊子所講的庖丁解牛、輪扁斫輪的故事有什麼區別呢？

圖九、「讀古詩詞網」介面

網站以單篇詩、詞或文章為單位，其中原文與譯文須分別從兩個不同網址擷取。為了一次擷取大量資料，我們一開始先由網站統整的類別找出大量文章連結，再分別擷取每一篇文章的原文網址，接著從原文網址中擷取譯文網址，開啟原文及譯文的網頁原始碼，擷取文字內容。最後分別將原文與譯文的文字內容存在兩個文字檔中，完成資料的收集。

實際作法如下：

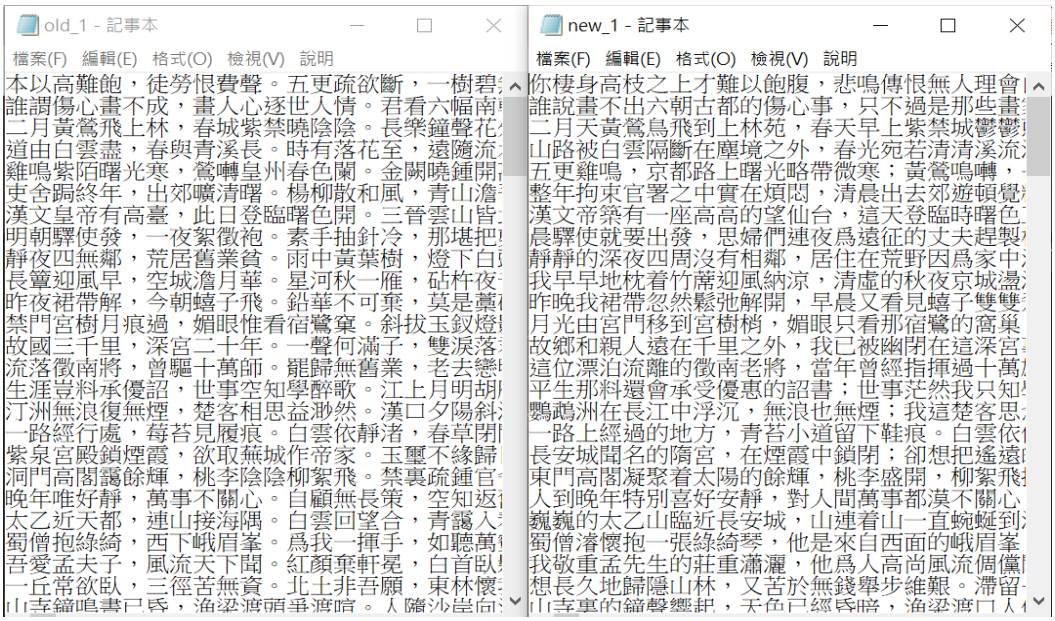


圖十、Web Crawler 運作流程圖

2. 資料處理和正規化

將資料以兩種方式分割：

(1) 以「篇」為單位分割資料，每行有一篇文章，作為一筆資料



圖十一、以「篇」為單位分割的資料處理結果

(2) 以「句」為單位分割資料，每行有一句文章，作為一筆資料

其步驟如下：

a. 以 Word 的「尋找與取代」功能，將逗號、句號、分號以換行符號取代

b. 比對古文與其譯文內容，確保該古文的行數對應到同一行的譯文

c. 以 Word 的「尋找與取代」功能，在字和字中間加上空格

圖十二、以「句」為單位分割的資料處理結果

最後將資料以 7:2:1 的比例分割為 train、valid、test 三份，即 train_古文、train_現代文 valid_古文、valid_現代文、test_古文、test_現代文，共六個 txt 檔。

這樣做的用意為在訓練模型時，分別將資料用於 Training、Validation 和 Testing。在進行 Training 時，我們會告訴模型標準答案，使模型能直接調整參數，再進行 Validation 來驗證，最後用 Testing 測試模型表現。

三種文體的資料量分別如以下二表所示：

	train set	validation set	test set	總計
唐詩	358	102	51	511
宋詞	219	63	31	313
散文	230	66	32	328
混合	807	231	114	1152

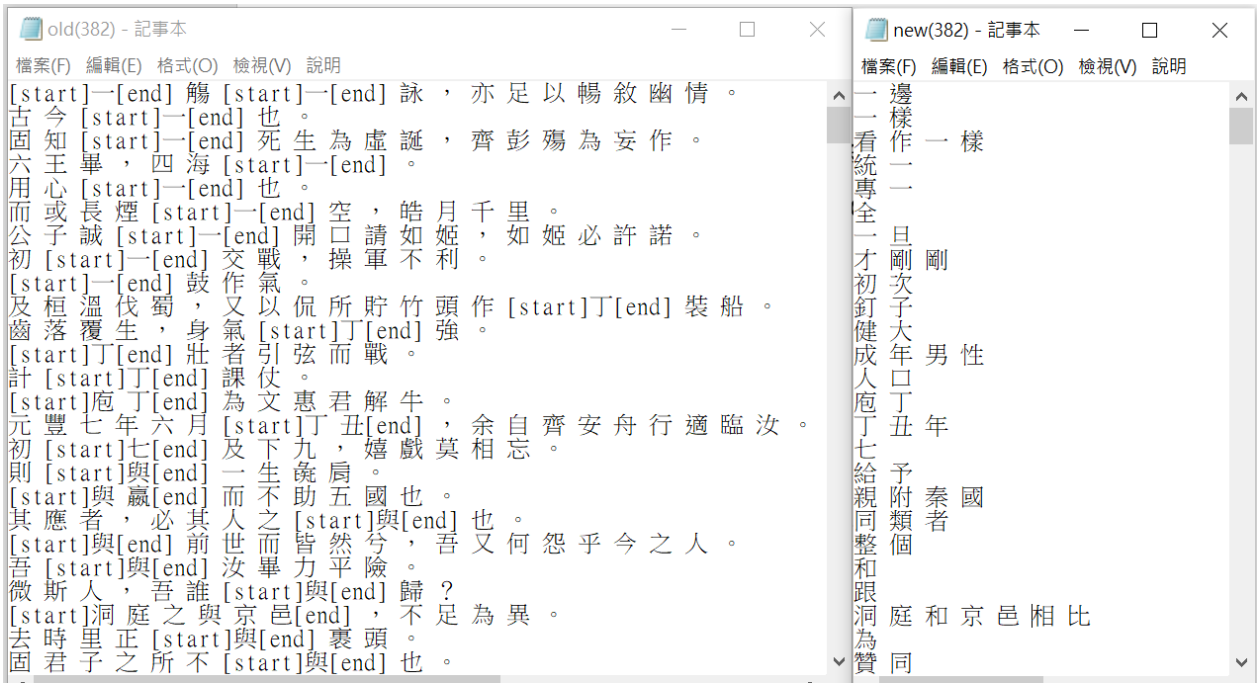
表一、以「篇」為單位分割出的資料量(單位：篇)

	train set	validation set	test set	總計
唐詩	1014	289	145	1448
宋詞	1824	521	260	2605
散文	5850	1671	837	8358
混合	8688	2481	1242	12411

表二、以「句」為單位分割出的資料量(單位：句)

(3) 將特定字詞的注釋以「用標籤提供單句前後文」的方式新增至 train set 中

我們從提供文言文詞語注釋的網站(https://wyw.hwxnet.com/)擷取大量詞語注釋，經過正規化處理後新增至 train set 中幫助訓練。目前完成正規化的共有 382 筆注釋。其資料格式如下：



圖十三、注釋資料格式

設計此格式的目的為希望能讓模型學習同一個字詞在不同前後文下的不同字義，同時又能讓模型專注於單一字詞的學習而非以整句作訓練。

(二)模型設計與訓練

1. Bert 模型

(1) 模型規格

我們使用的是 bert-base-chinese

a. hidden size : 768

b. number of attention heads : 12

(2) 模型設計

我們的輸入是 train_古文、train_現代文、valid_古文、valid_現代文、test_古文、test_現代文，共六個 txt 檔。

接著我們將以「選擇題」的方式訓練模型選出正確答案，形式為以古文為題目，對每一筆古文資料產生四個選項，其中一個選項為該筆古文資料對應到

的現代文資料(即該古文的翻譯)，作為正確選項，再從其他現代文資料中挑選三筆作為錯誤選項，其中錯誤選項的挑選方法會在下文詳細說明。

實際步驟如下：

a. 製作一份包含 Train_現代文、Valid_現代文、Test_現代文的集合 modern_all

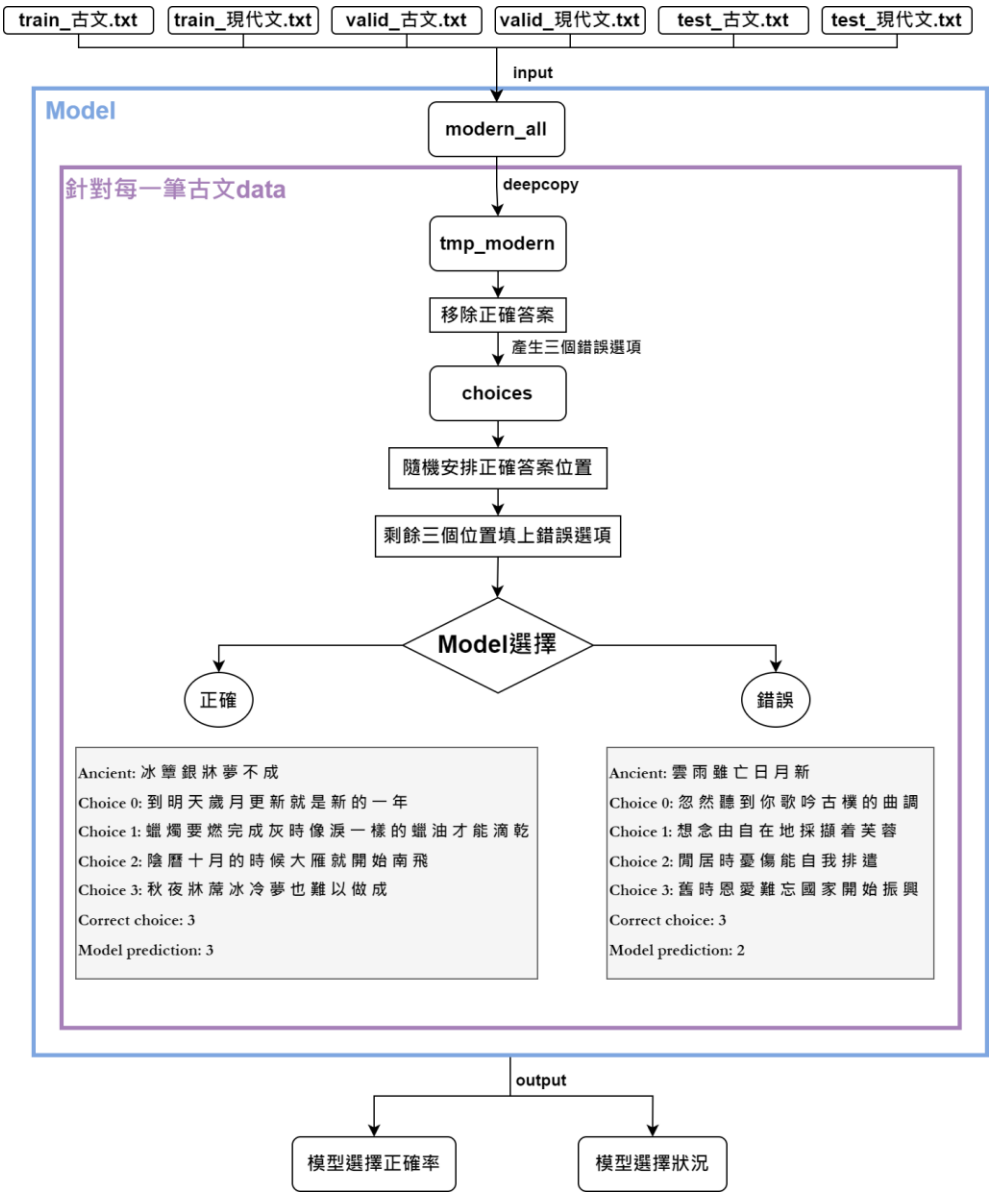
b. 在處理每一筆古文資料時，複製一份 modern_all 成為 tmp_modern

c. 從 tmp_modern 中將該筆古文資料將對應到的正確答案移除

d. 從 tmp_modern 中隨機挑選三個作為錯誤選項，放入 choices

e. 將正確答案隨機安排在四個選項的其中之一，其他三個位置則安排成錯誤選項，讓模型選擇

f. 輸出模型選擇正確答案的比率，以及模型的選擇狀況



圖十四、模型運作架構示意圖

(3) 錯誤選項的挑選方法

而三筆錯誤選項的挑選方法，我們嘗試了隨機選項與誘答選項兩種。

a. 隨機選項

起初，我們先嘗試了「隨機選項」的做法，即直接從 tmp_modern 中隨機選擇三個放入 choices 作為錯誤選項。

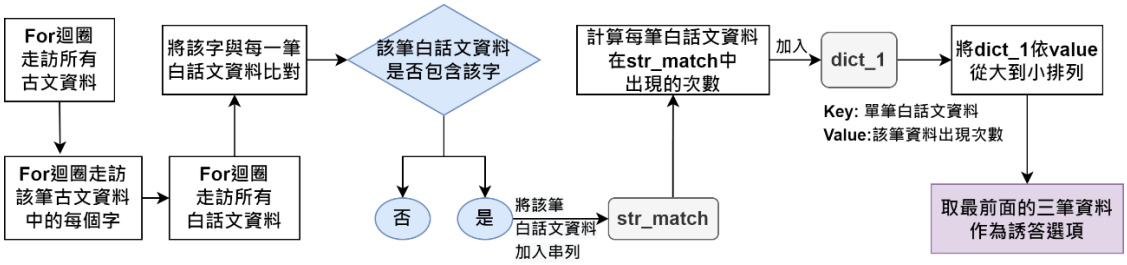
b. 誘答選項

經測試後我們發現，使用隨機選項的模型表現良好。因此，為了驗證模型的選擇能力，增加選擇時的難度，我們在選擇三個錯誤選項時，將隨機選擇改為選擇具誘答力的選項。

我們的做法為，建立一套「評估古文資料與現代文資料相似程度」的系統，在處理每一筆古文資料時，針對該筆古文資料評估每一筆現代文資料與它的相似程度。

挑選誘答選項的做法為，針對單筆古文資料，查看每一筆現代文資料與該古文的相似程度，取與該古文最相似的三筆現代文資料作為誘答選項。至於相似程度的評估標準，我們將之設定為「與該筆古文資料重複字數的多寡」。即對於每一筆現代文資料，若其中的字有越多是在該筆古文資料內出現過的字，即判斷它與該筆古文的相似程度越高。

針對每一筆古文資料的實際執行步驟如下：



圖十五、誘答選項挑選機制運作架構示意圖

(a) 利用 for 迴圈走訪該筆古文資料中的每個字

(b) 將每一筆現代文資料一一與該字比對，將有包含該字的現代文資料加入串列 str_match

(c) 計算每筆現代文資料在 str_match 中出現的次數，以字典 dict_1 的 key 儲存資料文字內容，value 儲存次數

(d) 將 dict_1 依 value 從大到小排列，取最前面的三筆資料作為選項

(4) 研究方法驗證

為了確認我們研究方法的適當性，我們需要確認「重複字最多的選項」與模型選擇的選項、正確答案是否皆有一定差異，進而判斷模型是否只是依照「選擇重複字最多的選項」的標準就能達成高正確率。

因此，我們作了以下測試：

a. 判斷「正確答案等於重複字最多的選項」的比率

b. 判斷「模型選擇選項等於重複字最多的選項」的比率

2. MT5 模型

(1) 模型規格

我們使用的是 Small size 的 MT5 模型

a. Number of layers : 8

b. hidden size : 512

c. number of attention heads : 6

(2) 模型設計

我們的輸入是 train_古文、train_現代文、valid_古文、valid_現代文、test_古文、test_現代文，共六個 txt 檔。

接著，我們將訓練模型把輸入的古文翻譯成對應的白話文後輸出。

經過 Training、Validation 和 Testing，輸出：

a. 翻譯結果

即模型翻譯「test_古文」的結果，以 txt 檔的形式輸出。

b. 模型效果評估

將模型翻譯「test_古文」的結果與「test_現代文」比對，輸出一個代表翻譯準確率的分數。

本研究中使用 BLEU 來評估模型的效果，BLEU 代表雙語評估替補，為 Bilingual Evaluation Understudy 的縮寫，是一套用來評估翻譯準確率的標準，結果與人類評估相近，亦可用於本研究「古文解譯」之成果評估。

BLEU score 越高，代表模型翻譯結果準確率越高。

伍、 研究結果

(一)選擇題

1. 我們嘗試了三種情況，分別是全部隨機選項、全部誘答選項和 train 用隨機，valid , test 用誘答選項，來看模型的正確率

		全部都用隨機選項	全部都用誘答選項	train 用隨機選項 valid , test 用誘答選項
唐詩	valid	99.31%	98.27%	91.00%
	test	100%	100%	95.86%
宋詞	valid	99.23%	94.82%	75.05%
	test	99.23%	96.92%	81.15%
散文	valid	98.79%	93.02%	85.43%
	test	96.05%	92.10%	80.24%

表三、三種文體在三種情況下的正確率比較

2. 我們算出「正確答案等於重複字最多的選項」和「模型選擇的等於重複字最多的選項」之比例，來確認我們研究方法的適當性

		正確答案等於重複字最多的選項	模型選擇的等於重複字最多的選項
唐詩	valid	92.39%	82.01%
	test	94.48%	95.55%
宋詞	valid	79.85%	69.10%
	test	85.77%	73.08%
散文	valid	45.07%	27.31%
	test	51.06%	30.40%

表四、兩項驗證結果的比例

(二)翻譯

1. 利用不同的 train set 跟 test set 來測試何種準確率最高，mix 代表唐詩+宋詞+散文

train - test	BLEU score
唐詩 - 唐詩	8.8011
宋詞 - 宋詞	9.3029
散文- 散文	8.1069
mix - 唐詩	7.0841
mix - 宋詞	8.9797
mix - 散文	8.1441

表五、六種情況的翻譯準確率

2. 將三種文體資料量統一為 1448 比前後之準確率比較

train - test	統一資料量前 BLEU score	統一資料量後 BLEU score
唐詩 - 唐詩	8.8011	8.2637
宋詞 - 宋詞	9.3029	9.9372
散文- 散文	8.1069	7.9941

表六、統一資料量前後之準確率比較

3. 以單句訓練與以整篇訓練之準確率比較

	單句 BLEU score	整篇 BLEU score
唐詩	8.8011	0.1883
宋詞	9.3029	0.1632
散文	8.1069	0.2005

表七、單句與整篇之準確率比較

4. 新增注釋資料前後之準確率比較 (train - test 皆使用三種文體混合的 mix set)

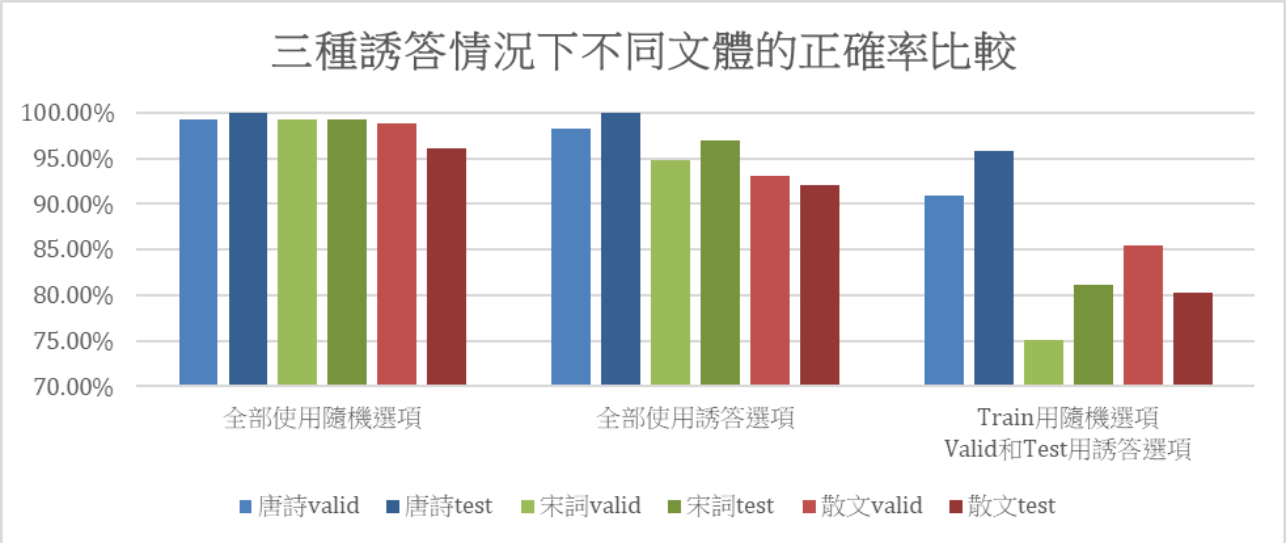
	BLEU score
新增注釋前	9.2333
新增注釋後	9.5696

表八、新增注釋資料前後之準確率比較

陸、討論

一、選擇題

(一)不同文體的正確率差別



圖十六、三種誘答情況下不同文體的正確率比較

在實驗中，我們將全部用隨機選項的情況判定為難度最低；全部用誘答選項的情況難度為次低；而 Train 用隨機，Valid 和 Test 用誘答選項時，因用較簡單的選項訓練，但用較困難的選項驗證及測試，故我們判斷此情況為難度最高。

將表一繪製成圖可發現，在全部用隨機選項的情況下，唐詩、宋詞、散文三者的正確率皆接近 100%，差異不大，但仍呈現唐詩稍高、散文稍低的趨勢；在全部用誘答選項的情況下，正確率由高至低依序為唐詩、宋詞、散文；在 train 用隨機選項，valid 和 test 用誘答選項的情況下，唐詩與散文的正確率相差不大，而宋詞明顯較低。由整體趨勢來看，唐詩的正確率稍高於其他兩者，因此我們推測宋詞與散文的難度應較唐詩高，但整體相差不大。

另外，若從單一文體的角度檢視，三種文體的正確率皆隨我們預期的難度增加而呈現稍微下降的趨勢，表示誘答確實能增加模型選擇的難度。其中又以宋詞正確率下降的幅度最大，表示模型在宋詞的表現受到誘答與否的影響較大。

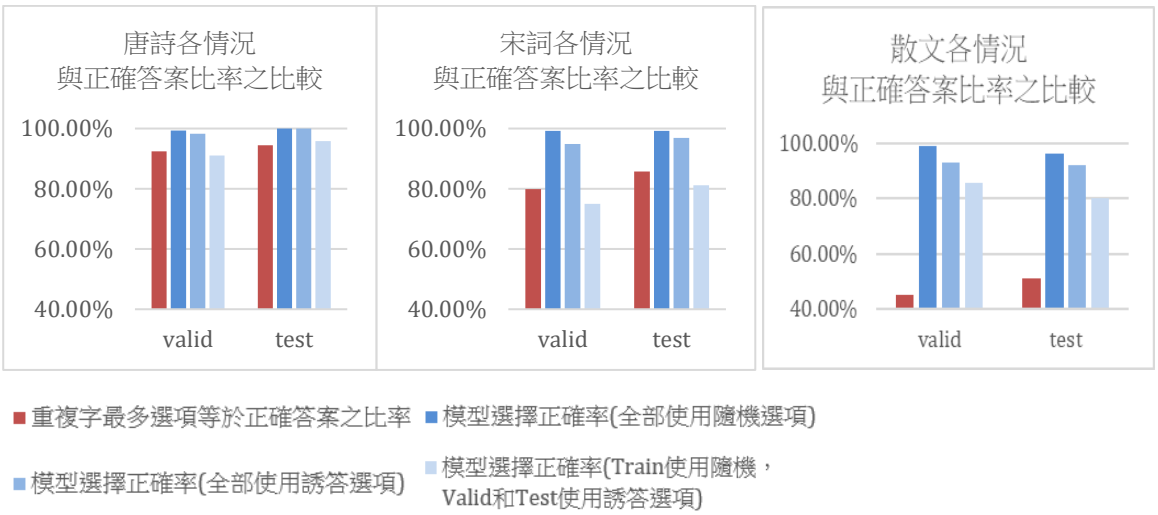
(二)模型訓練方法之驗證

1. 「重複字最多選項等於正確答案之比率」與「模型選擇正確率」的比較

		重複字最多選項等於正確答案之比率	模型選擇正確率		
			全部使用隨機選項	全部使用誘答選項	Train 使用隨機, Valid 和 Test 使用誘答選項
唐詩	valid	92.39%	99.31%	98.27%	91.00%
	test	94.48%	100%	100%	95.86%
宋詞	valid	79.85%	99.23%	94.82%	75.05%
	test	85.77%	99.23%	96.92%	81.15%
散文	valid	45.07%	98.79%	93.02%	85.43%
	test	51.06%	96.05%	92.10%	80.24%

表九、「不同情況下的模型選擇正確率」與「重複字最多選項等於正確答案之比率」比較(內容結合表三、表四左半部分)

將唐詩、宋詞、散文分別作圖：



圖十七、三種文體之「不同情況下的模型選擇正確率」與「重複字最多選項等於正確答案之比率」

由圖表可見，在三種文體中，「全部使用隨機選項與全部使用誘答選項之模型選擇正確率」皆高於「重複字最多選項等於正確答案之比率」，表示模型不是依照重複字多寡去選擇，此訓練方法具有成效。

而在「Train 使用隨機，Valid 和 Test 使用誘答選項之模型選擇正確率」與「重複字最多選項等於正確答案之比率」的比較中，我們發現：

對於唐詩和宋詞，前者皆略低於後者，可推測模型在只有用隨機選項作訓練時，可能會傾向於選擇與古文重複字較多的選項，因此在使用誘答選項作驗證與測試時，模型表現得較差。

對於散文則推測是因資料本身「重複字最多選項等於正確答案之比率」較另外兩種文體低許多，故無顯現出此趨勢。

2. 「模型選擇的選項」等於「重複字最多的選項」的比率

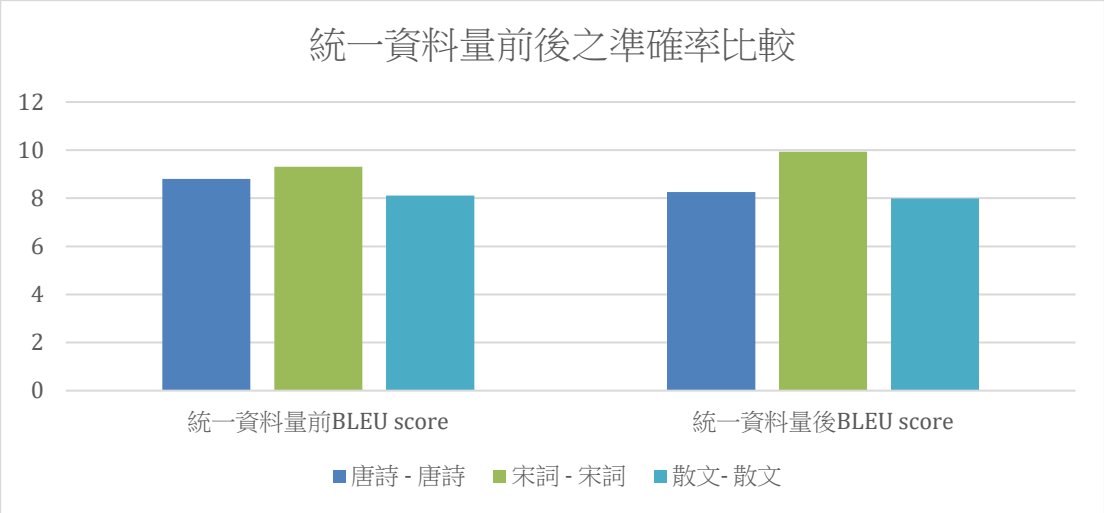
		模型選擇的等於重複字最多的選項的比率
唐詩	valid	82.01%
	test	95.55%
宋詞	valid	69.10%
	test	73.08%
散文	valid	27.31%
	test	30.40%

表十、「模型選擇的選項」等於「重複字最多的選項」的比率(內容同表四右半部分)

除唐詩 test 以外，其餘數值皆不高於 90%，可推測模型非依照選擇重複字較多的標準去選擇，應有進一步的判定標準。

二、翻譯

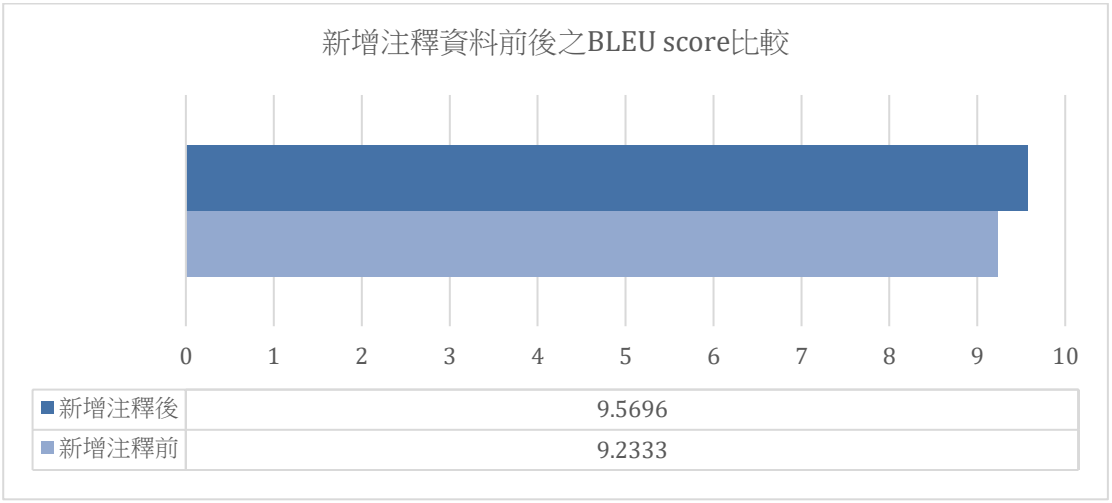
(一)統一資料量前後三種文體準確率趨勢比較



圖十八、統一資料量前後三種文體準確率比較

將統一資料量前後三種文體之準確率分別作圖可見，統一資料量前後之 BLEU score 皆為宋詞最高，唐詩其次，散文最低。並且三種文體統一資料量前後之 BLEU score 差異不大，我們推論統一資料量對三種文體分別訓練之準確率影響不大。

(二)新增注釋資料前後翻譯準確率比較



圖十九、新增注釋資料前後翻譯準確率比較

由圖可知在新增注釋後，準確率有些微上升，雖然因為新增注釋的資料量較少，故準確率上升幅度不大，但以新增少量注釋就讓準確率有所上升這點來看，我們認為新增注釋是能有效提升模型準確率的研究方向。未來我們將持續完成注釋資料的正規化，若成功完成所有注釋資料的正規化，注釋資料量可達到約 20000 筆。

(三)機器實際翻譯結果

從表三可知六種情況準確率相差不大，因此我們決定挑出幾句機器實際翻譯的結果來看整體模型狀況：

1. 代名詞翻譯情況

模型對部份代名詞仍常有翻譯錯誤之情形，舉例如下：

(1) 翻譯正確的

a. 見爾當何秋

機器翻譯：見到你的秋秋

(將「爾」正確翻成「你」)

b. 此心安處是吾鄉

機器翻譯：此心安處是我鄉

(將「吾」正確翻成「我」)

c. 角者吾知其爲牛

機器翻譯：我知道它是牛

(將「吾」正確翻成「我」、將「其」正確翻成「它」)

(2) 翻譯錯誤的

a. 問之不肯道姓名

機器翻譯：問你不肯道姓名

(「之」在此指「他」，翻錯成「你」)

b. 其餘以汝飽

機器翻譯：這些餘都是我的飽飽

(「汝」原意為「你」，翻錯成「我」)

2. 輸入句子的長短與翻譯表現關係

句子短的翻譯情況較好，舉例如下

(1) 句子長

a. 揹負青天而莫之夭閼者

機器翻譯：只要揹負青天而不夭夭

(原意：揹負青天而沒有什麼力量能夠阻遏它了)

b. 累累如人之旅行於牆外而見其髻也

機器翻譯：累累如人在牆外看見他的髻子

(原意：山峰重重疊疊的樣子正如有人在牆外行走而看見的那人髮髻的形狀一樣)

c. 無高山大陵曠途絕險爲之關隔也

機器翻譯：不是高山大陵曠途絕險爲他的關隔

(原意：不用廣闊險峻的高山土丘就能把它困住)

(2) 句子短

a. 依稀記

機器翻譯：依稀記得

b. 思往事

機器翻譯：思念往事

c. 且吾聞之

機器翻譯：而且我聽說這樣

d. 何者

機器翻譯：爲什麼呢

e. 當此之時

機器翻譯：當這時

3. 其餘翻得較好的句子

(1) 可憐王孫泣路隅

機器翻譯：可憐王孫哭泣路旁

(2) 初聞涕淚滿衣裳

機器翻譯：初聽到涕淚滿滿衣裳

(3) 文采風流今尚存

機器翻譯：文采的風流今日尚存

(4) 底事崑崙傾砥柱

機器翻譯：崑崙山上傾斜的砥柱

(5) 春無蹤跡誰知

機器翻譯：春天無蹤跡誰知道

(6) 但莫管春寒

機器翻譯：但卻不管春寒

(7) 臣未嘗聞也

機器翻譯：臣子沒有聽說

(8) 此圯上老人所爲深惜

機器翻譯：這樣的人們是老人所爲深惜的

(9) 而其狀貌乃如婦人女子

機器翻譯：而其形貌是像婦人女子

4. 翻不好的

(1) 翻譯錯誤的

a. 昔日長城戰

機器翻譯：今日長城戰戰

(「昔日」翻錯成「今日」)

b. 天下有大勇者

機器翻譯：天下有大偉大的人

(「勇者」翻錯成「偉大的人」)

c. 其爲形也不類

機器翻譯：這樣的形形不可以

(「不類」原意為「不像」，翻錯成「不可以」)

d. 若麟之出

機器翻譯：像麟子出來

(「若」在此指「如果」，翻錯成「像」)

e. 如在野

機器翻譯：如果在野裏

(「如」在此指「像」，翻錯成「如果」)

(2) 沒有翻出來的

a. 則謂之不祥也亦宜

機器翻譯：就是說不祥的

(「亦宜」意思是「也是合適的」，此處未翻出)

b. 不爲不加矣

機器翻譯：不是不加了

(「加」在此指「厚愛有加」，此處未翻出)

c. 仕不爲貧

機器翻譯：不是貧窮

(原意為「爲 官 不 是 因 爲 貧 窮」，「爲 官」、「因 爲」皆未翻出)

(3) 照抄古文的

a. 判司卑官不堪說

機器翻譯：判司卑官不堪說

b. 蟬鳴空桑林

機器翻譯：蟬鳴空桑林

c. 伏屍百萬

機器翻譯: 伏屍百萬

(4) 相同字重複多次

a. 而卒賴其力

機器翻譯：而且不能夠夠夠夠夠他們的力量

b. 對瀟瀟暮雨灑江天

機器翻譯：對瀟瀟瀟的瀟瀟的雨灑在江天

5. 「主觀判定較接近白話文之用詞的古文輸入」與「較不接近白話文用詞之古文輸入」的翻譯情況

當輸入之古文較接近白話文之用詞時，翻譯結果較佳。

(1) 比較接近白話文用詞的

a. 忽然遭世變

機器翻譯：忽然遭遇世變

b. 波撼岳陽城

機器翻譯：波濤震撼岳陽城

c. 倚杖聽江聲

機器翻譯：倚杖聽着江水聲

d. 夢繞神州路

機器翻譯：夢中繞着神州路

e. 古之所謂豪傑之士者

機器翻譯：古代的豪傑之士

f. 以匹夫之力

機器翻譯：以匹夫的力量

(2) 比較不接近白話文用詞的

a. 亦免冠徒跣

機器翻譯：也免冠徒的冠冕

(原意：也不過就是摘掉帽子光著腳)

b. 海運則將徙於南冥

機器翻譯：海運就會遷徙在南冥

(原意：隨著海上洶涌的波濤遷徙到南方的大海)

c. 南冥者

機器翻譯：南冥人

(原意：南方的大海)

6. 字義密度(即古文一字大約對應到白話文幾個字) 與翻譯情況

字義密度過大或過小，翻譯效果皆不佳。

(1) 字義密度大

a. 至人無己

原意：真實自然的人沒有自我的偏見

機器翻譯：到人不有己

b. 而不加勸

原意：他卻並不會因此而更加奮勉

機器翻譯：而不加勸

(2) 字義密度小

a. 亦若此矣

原意：如此而已

機器翻譯：亦若此矣

b. 世皆稱

原意：世人都稱

機器翻譯：世界都稱讚

7. 新增注釋後翻譯變好的句子

a. 「然」麟之爲物

原意：「但是」麟是野生動物

新增注釋前機器翻譯：「那麼」麟是個物

新增注釋後機器翻譯：「可是」麟是個物

(將「然」正確翻成「可是」)

b. 「行」比一鄉

原意：「行爲」可以順應一鄉羣衆

新增注釋前機器翻譯：「行走」比一鄉

新增注釋後機器翻譯：「行爲」比一個鄉

(將「行」正確翻成「行為」)

c. 「月明」多被雲妨

原意：「月亮雖明」卻總被雲遮住

新增注釋前機器翻譯：「月明」多被雲阻礙

新增注釋後機器翻譯：「月亮明亮」多被雲阻礙

(將「月明」更精確詳細地翻出了「月亮明亮」)

柒、結論

一、以預訓練的語言模型做古文的文意理解

(一)模型在選擇題的狀況表現非常好，比只選重複字好很多。

(二)宋詞與散文的難度應較唐詩稍高，但相差不大。

(三)大部份資料在使用重複字較多的選項作誘答時，相較於使用隨機選項時，模型準確率較低。

(四)模型應有一定選擇能力，其選擇之標準並非單單依照重複字多寡去判定。

二、以 MT5 作翻譯訓練之成效

(一)模型對唐詩、宋詞、散文之翻譯準確率差異不大。

(二)模型翻譯較佳的古文輸入情況有：

1. 句子較短
2. 用詞較接近白話文
3. 字義密度適中

(三)模型翻譯較差的情況有：

1. 句子較長
2. 用詞較不接近白話文
3. 字義密度極大或極小
4. 句子包含代名詞

(四)新增注釋後，模型翻譯準確率有小幅度上升。

(五)本研究較大的難關是人工正規化的過度繁瑣，導致我們無法有效率地將爬蟲取得的所有訓練用資料都完成正規化。目前的發展方向為開發新的半自動或全自動的正規化程序，期望能將剩餘約 40000 筆的資料完成正規化以擴大訓練集，增加模型翻譯準確率。最終目標是希望能將訓練完成的模型製成網頁或應用程式供大眾使用，並將模型推廣至多語言的古文解譯。

捌、參考資料及其他

[1] *创新实训(12)-生成式文本摘要之 T5_ttxs69 的博客*. (2020, July 1). CSDN 博客. Retrieved January 20, 2022, from https://blog.csdn.net/qq_34842847/article/details/107066613

[2] *config.json · google/mt5-small at main*. (2020, November 15). Hugging Face. Retrieved February 3, 2022, from https://huggingface.co/google/mt5-small/blob/main/config.json

[3] Goodfellow, I. (2016). Sequence Modeling: Recurrent and Recursive Nets. In *Deep Learning*. MIT Press. https://www.deeplearningbook.org/contents/rnn.html

[4] Khetan, A., & Karnin, Z. (n.d.). *schuBERT: Optimizing Elements of BERT*. ACL Anthology. Retrieved January 5, 2022, from https://aclanthology.org/2020.acl-main.250.pdf

[5] *Language Modeling with nn.Transformer and TorchText — PyTorch Tutorials 1.10.1+cu102 documentation*. (n.d.). PyTorch. Retrieved February 27, 2022, from https://pytorch.org/tutorials/beginner/transformer_tutorial.html

[6] Lee, H. (2019). *Transformer*. https://youtu.be/ugWDIIOHtPA

[7] Min. (2021, 1 21). *[筆記] Attention 及 Transformer 架構理解*. Retrieved January 5, 2022, from https://mkh800.medium.com/%E7%AD%86%E8%A8%98-attention-%E5%8F%8A-transformer-%E6%9E%B6%E6%A7%8B%E7%90%86%E8%A7%A3-c9c5479fdc8a

[8] Nabi, J. (2019, July 11). *Recurrent Neural Networks (RNNs). Implementing an RNN from scratch in… | by Javaid Nabi*. Towards Data Science. Retrieved December 27, 2021, from https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85

[9] *Pretrained models — transformers 2.5.1 documentation*. (n.d.). Hugging Face. Retrieved Feburary 3, 2022, from https://huggingface.co/transformers/v2.5.1/pretrained_models.html

[10] *自監督學習 self-supervised learning 介紹*. (2021, June 11). 藏字閣. Retrieved February 27, 2022, from https://writings.jigfopsda.com/zh/posts/2021/self_supervised_learning/

[11] *從零開始的 Sequence to Sequence*. (2017, September 28). 雷德麥的藏書閣. Retrieved February 3, 2022, from http://zake7749.github.io/2017/09/28/Sequence-to-Sequence-tutorial/

[12] Ting. (n.d.). *Transformer 李宏毅深度學習*. HackMD. Retrieved January 20, 2022, from https://hackmd.io/@abliu/BkXmzDBmr

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (n.d.). Attention is all you need. https://arxiv.org/abs/1706.03762

[14] Andy. (2019, October 31). T5 模型：NLP Text-to-Text 預訓練模型超大規模探索- CodingNote.cc. - CodingNote.cc. Retrieved January 16, 2022, from https://codingnote.cc/zh-tw/p/19594/

[15] 邱冠龍. (2021, 12 03). *自然語言處理預訓練模型之 T5: Text-to-text transfer Transformer 簡介*. FIND. Retrieved February 16, 2022, from https://www.find.org.tw/index/wind/browse/e26c9d9800df05e7a1e372a779a2217d/

[16] (n.d.). 【古詩詞大全】 古詩詞名句古詩詞、文言文翻譯及賞析讀古詩詞網-dugushici. Retrieved January 03, 2022, from https://fanti.dugushici.com/

[17] (n.d.). 漢語網. Retrieved December 22, 2021, from https://www.chinesewords.org/

【評語】190012

很好的嘗試，請持續加油。建議先以 Google Translation 服務的方式翻譯唐詩為主，待成效良好時，再陸續擴充至其他領域。

此作品利用深度學習創建一個將輸入的古文翻譯成白話文的系統，其目前評估翻譯正確率的方式採用選擇題的方式，測試時所用選項的產生則有隨機選項和誘答選項兩種方式。古文的選擇則有唐詩、宋詞和散文的種類。作品中有探討不同文體的正確率，目前測試所得的正確率有 90%左右，但這是因為是採用選擇題的方式來進行正確率評估。作品中有進行「重複字最多選項等於正確答案之比率」與「模型選擇正確率」的比較，此外也有直接看翻譯出來的字句是否合理(但這部分判斷較為主觀)。此作品是屬於語言翻譯的領域，未來在效能評估上可探討如何使用 Social Media 讓大眾參與評審進而提升翻譯的正確性，得到更客觀且令人信服的評比方法。