二、研究計畫內容

(一) 摘要

基於深度學習模型之問句生成(Question Generation)技術近年來受到許多研究者的注目，也有許多卓越的研究成果。目前問題生成之技術主要是基於一給定之短文(Context Paragraph)與答案片段(Answer)生成對應之問題。問句生成技術最直接的實際應用為：教育學習產業中之閱讀測驗自動化出題。我們可以利用問句生成技術可以針對任意文章任意指定片段來生成個人化閱讀測驗。然而，目前之生成技術乍看可行，但仍離自動化出題有一段距離。其中一個可改善的問題為：問句生成模型仍須仰賴使用者給定答案片段，為此使用者必須先行閱讀過文章始可進行答案之挑選，此部分仍須人工介入。我們認為一個比較理想的方式為僅輸入 Context Paragraph 就好。也就是我們輸入一篇文章後，系統便自動找尋合適之答案片段，並將萃取之答案片段，輸入現有問句生成模型，產生對應之問句。本研究中，我們預計研究如何從一篇輸入的文章中，透過深度學習方式自動化找尋合適成為考題答案之片段(我們稱之為考眼)，來進一步改善現有閱讀測驗自動化出題仍須由仍人工指定答案片段之問題。

(二) 研究動機與研究問題

近年來問句生成(Question Generation)技術之逐漸成熟[10][11]，吸引眾多研究者與商業應用者之投入。使得自動問題生成成為開發未來學習系統理想的工具。本計畫希望利用自然語言處理技術，改善如今自動問句生成系統的問題，讓自動問句生成系統能更完善，讓自動問句生成技術能更貼近真實的人工出題品質。如今，自動問句生成之閱讀理解模型已廣泛應於多種教育用途，像是學術寫作支持(academic writing support)[1]、閱讀理解評估(reading comprehension assessment)[2]、教育聊天機器人(educational chatbots)[3]。

目前問題生成之技術主要是基於一給定之短文(Context Paragraph)與答案片段(Answer)生成對應之問題。以下我們以中興大學范耀中老師研究團隊所開發之問句生成系統(可於 http://app.queratorai.com/實際使用)為例說明現階段問句生成之應用模式與可達成的品質。以下輸入之 Context Paragraph 為一篇即時新聞，紅色標記部分為我們人工所挑選之答案。我們可以從下列圖表看到，現有系統所生成之問題已達流暢品質並且語意正確。此外，目前中英文之生成皆可有一定程度之效果，並且進行多樣化的生成(同樣一個答案可以有多種不同的問法)。

Input Context Paragraph
新冠肺炎疫情持續，陽明交通大學今天舉行合校後第一場實體記者會，該校生物科技學院院長楊進木發表研究成果，團隊用 AI 找出一種抗發炎舊藥 JMY206 能抗新冠肺炎，經動物實驗證實，藥效比瑞德西韋強約 30 倍，為疫情帶來一線曙光。 陽明交大校長林奇宏表示，這項研究在合校前就已啟動，團隊透過 AI，從全球已在使用的大量藥物中，篩出抗發炎舊藥 JMY206，還有巴色匹韋（Boceprevir）、特拉匹韋（Telaprevir）、奈非那韋（Nelfinavir）共四種舊藥，有望治療新冠肺炎。老藥新用可降低資金，上市前的臨床試驗時間也比新藥少一到數年。這將是台灣搶占全球抗疫領先地位的最佳策略。 楊進木表示，大數據平台，比對新冠病毒與 SARS 等病毒，發現相似度大於 95%。團隊透過 AI，模擬宿主受病毒感染後的體內細胞變化機制，並比對全球超過 250 份已公開的冠狀病毒主要蛋白酶 3D 構造圖像，發現新冠病毒侵入人體細胞的關鍵蛋白質具有六種可能的動態結構，鎖定蛋白酶的單門，在三個月內找到具有抑制病毒活性效果的三種潛力藥物，包括巴色匹韋（Boceprevir）、特拉匹韋（Telaprevir）、奈非那韋（Nelfinavir），已發表於國際期刊 ACS Nano。 另外，楊進木表示，團隊發現一款抗發炎舊藥 JMY206，經細胞實驗發現，它抑制新冠病毒效果比國際目前期待的新藥「瑞德西韋」強約 30 倍，動物實驗也證實有治療效果，是有潛力的口服藥物。團隊正在試著申請專利。 楊進木表示，這次的研發團隊陣容堅強，除陽明交大，還有台灣大學、中研院、中興大學和設有國內唯一 P4 等級病毒實驗室的國防醫學院預防醫學所，研發主力包括陽明交大博士生 Nikhil Pathak 倪齊歐、許彥超、許農育和陳筠媞。

Answer	Generated Question
楊進木	Q1: 哪個人擔任陽明交通大學生物科技學院院長?. Q2：誰的研究成果為新冠肺炎疫情帶來一線曙光?
三種潛力藥物	Q1：巴色匹韋和特拉匹韋都屬於哪一種藥物? Q2：巴色匹韋屬於哪一種藥物?
相似度大於 95%	Q1：楊進木表示，大數據平台比對新冠病毒與 sars 等病毒，發現什麼?

問句生成技術最直接的實際應用為：教育學習產業中之閱讀測驗自動化出題。我們可以利用問句生成技術可以針對任意文章任意指定片段來生成個人化閱讀測驗。然而，目前之生成技術乍看可行，但仍離自動化出題有一段距離。其中一個可改善的問題為：問句生成模型仍須仰賴使用者給定答案片段，為此使用者必須先行閱讀過文章始可進行答案之挑選，此部分仍須人工介入。如同上例中所述，三個紅色的答案片段為我們所提供之輸入之一(另一輸入為 Context Paragraph 本身)。

我們認為一個比較理想的方式為僅輸入 Context Paragraph 就好。也就是我們輸入一篇文章後，系統便自動找尋合適之答案片段，並將萃取之答案片段，輸入現有問句生成模型，產生對應之問句。也就是說我們想只藉由輸入 Context Paragraph C 透過一個深度學習模型 M_A 進行考眼之萃取

$M_A(C) \rightarrow \{A_1, ..., A_m\}$

取得$\{A_1, ..., A_m\}$之後再透過現有問句生成模型 M_{QG} 來生成問句

For all A_i :

$M_{QG}(C, A_i) \rightarrow Q_i$

可能解決方案 ：

針對上述的目標，一個簡單的想法為使用 NER (Name Entity Recogonition) 技術及現有 NLP Toolkit 如 NLTK[8]、spaCy[9]，但他們就僅能針對文章中的人名、地點、時間...等，進行標記，但我們知道如果要完成一個完整的自動出題系統，並能確保學生真的完整理解一篇文章的內容，我們不應該將考題侷限於人名、地點...等較短專有名詞上。因為上面提到的工具關鍵字挑選的方式較簡單，較難挑選出一段文字作為答案，造成在問句生成時，只能針對片段的知識點，進行考題的生成。相較之下，人工出題較喜歡針對文章中較廣較長的知識點，作為出題的方向，來檢驗學生是否真的完全了解文章的語意。因此，現有的工具，因為關鍵字的挑選太粗淺，造成產生的問題較沒有鑑識度，無法真正評估學生在學習上的效果，這也是現在自動問題生成系統還無法取代人工出題的其中一項原因。

為了改善現有關鍵字選取只能侷限於單詞的選擇，本計畫想開發一套能篩選出閱讀文本中的片段做為答案的系統，欲透過機器學習來達到增長關鍵字的選擇為一段文字，來提升整體自動問句生成系統的品質。如此不僅能增進自動問句生成系統的問題難度，更能深入了解學生是否真的完全了解此篇文章中的內容。

以圖一為例，Text 1 的內容為美國布希總統在選舉時，請到了美國著名的民調公司 Gallup 來為自己做民調，而民調的時間是從 1992 年 11 月至 1993 年 1 月之間製作的，但我們使用 spaCy 這款套件來做 NER，我們發現雖然 spaCy 為我們分別挑出了「November 1992」和「January

1993」兩個關鍵字，也就是 spaCy 認為這兩個關鍵字為獨立的個體，但以語意上來說，「November 1992 through January 1993」表達的是一段時間，而並非兩個時間點，如果將兩個時間點分割成兩個獨立的個體，會讓學生誤以為這兩個時間並不存在連貫性，因此這樣的斷詞錯誤會造成學生在閱讀理解上的錯誤。

圖一：

Text 1 : George H.W. Bush averaged an 84% approval rating among Republicans in an average of November 1992 through January 1993 Gallup polls.
Result : November 1992 (DATE)
January 1993 (DATE)

因此本計畫預計透過 BERT[6]模型來解決上述的問題。BERT 是一種可微調的語言模型(Language Model,LM)，語言模型可以在給定一些詞彙的前提下，估計下一個詞彙出現的機率，而傳統語言模型會針對不同 NLP 任務去設計一個最適合的神經網路架構，但設計集測試的過程中會耗費許多人力、時間、計算資源。BERT 模型就是一個是先訓練好，而且可以套用到多個 NLP 任務的模型，而我們再以此架構去微調，達成我們想完成的任務。

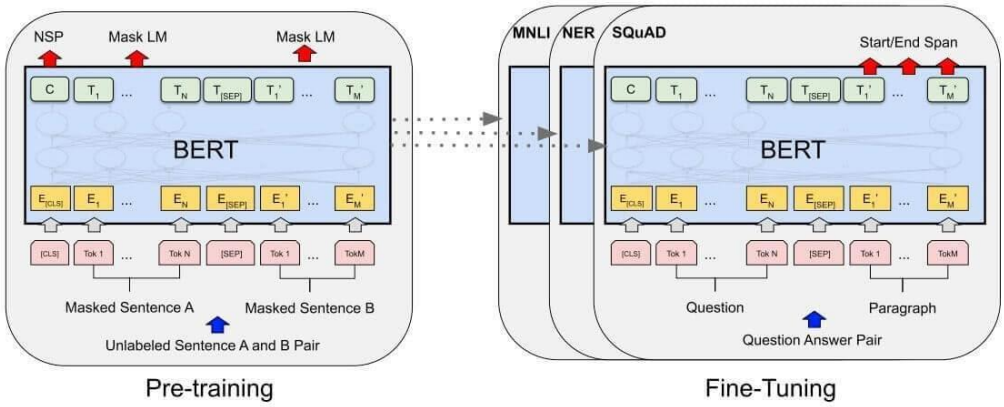
(三) 文獻回顧與探討

我們會在此部分針對計畫中會使用到的 SQuAD 資料集[5]、BERT 模型，進行介紹並談論使用它們的原因。

1. SQuAD(The Stanford Question Answering Dataset)[5] : SQuAD 資料集從 Wikipedia 中選擇前一萬篇熱門的文章，並隨機選擇了 500 多篇文章，並抽取出兩萬多個段落並且覆蓋了各式各類的主題。團隊雇用了 Mturk[1] Worker ，要求 Mturk Worker 讀一個段落並針對這個斷落的內容提最多 5 個問題，同時在段落中標註出對應答案的位置，並針對別人提出的問題，在原文中用盡量少的字長標註出答案所在。SQuAD 資料集除了資料量大之外，與之前同類型的資料集個重大區別及回答不是選擇題，而是問答題。而且 SQuAD 的答案是在單一個段落裡，與其他資料集會跨文件有很大的不同。

[1] Amazon Mechanical Turk (MTurk) is a marketplace for completion of virtual tasks that requires human intelligence. The Mechanical Turk service gives businesses access to a diverse, on-demand, scalable workforce and gives Workers a selection of thousands of tasks to complete whenever it's convenient.

2. BERT 模型(Bidirectional Encoder Representations from Transformers) : BERT 是一種可微調的語言模型(Language Model,LM)，語言模型可以在給定一些詞彙的前提下，估計下一個詞彙出現的機率，而傳統語言模型會針對不同 NLP 任務去設計一個最適合的神經網路架構，但設計集測試的過程中會耗費許多人力、時間、計算資源。因此就有 BERT 模型架構的誕生，BERT 模型就是一個是先訓練好，而且可以套用到多個 NLP 任務的模型，而我們再以此架構去微調，達成我們想完成的任務。



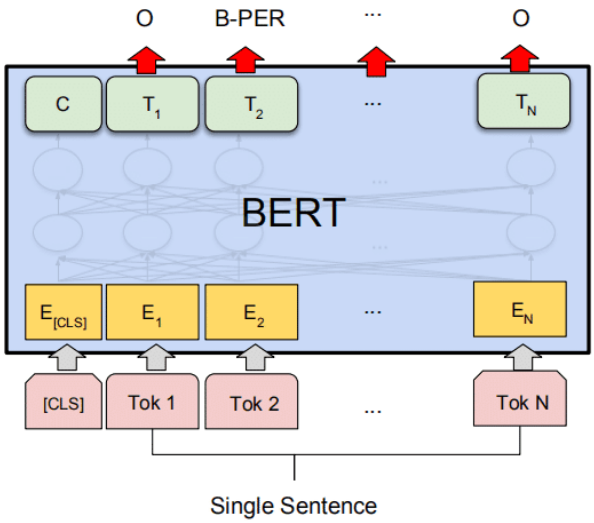
圖二 : Overall pre-training and fine-tuning procedures for BERT.

(四) 研究方法與步驟

本計畫預計使用 SQuAD 作為訓練資料，並使用 BERT 預訓練模型進行微調，來提升自動問句生成系統中的關鍵字選擇。而我們選擇 SQuAD 資料集的原因則是因為 SQuAD 是市面上第一個 span-based(指問題的答案是從文章中標註出來的)的資料集，雖然 span-based 的特性同時限制了問題及答案的多樣性，但卻也使的問題的答案被限制在文章中。此外，SQuAD 資料集克服了市面上現存的閱讀理解資料集的兩個問題(i)品質較高的資料集通常資料太小，無法訓練出有效的資料模型(ii)資料量大的資料集通常是半合成的，因此與平時所閱讀的閱讀理解題組不具有相同的特徵。也就是 SQuAD 資料集是現在資料量較大且兼顧品質的閱讀理解資料集。

1. 在 BERT 之上加入新的線性分類器：

我們在(三)文獻回顧與探討中所提及的 BERT 是一個可以微調的語言模型，也就是我們可以在預訓練完的 BERT 之上加入新的線性分類器(Linear Classifier)，並利用下游任務的目標函式從頭訓練分類器並微調 BERT 的參數。而我們預計使用 HuggingFace[7]團隊以訓練的 PyTorch Fine-tuning BERT 為基底，並搭配 transformer 模型庫中的 BertForTokenClassification 套件。BertForTokenClassification 是一個包含了 BERT 模型並加上字詞標註的分類器的可微調模型。



圖三: BertForTokenClassification

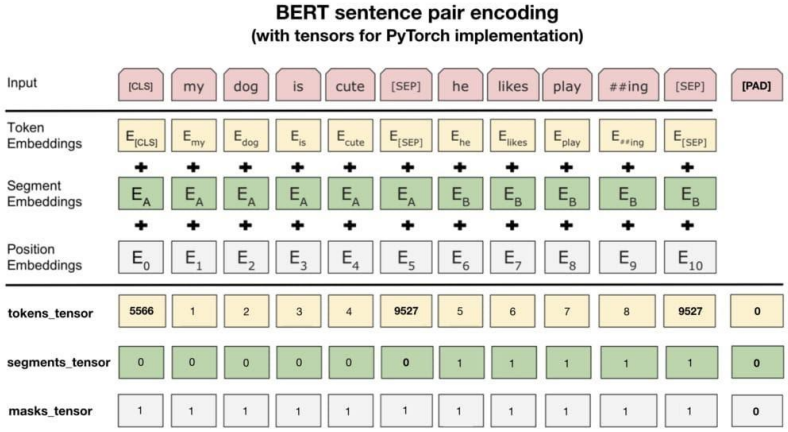
2. 將 SQuAD 資料集轉換成 BERT 相容的格式：

```
structure:
  article (dict)
  ├── title (str)
  ├── paragraphs (dict)
      ├── context (str)
      └── qas (dict)
          ├── question (str)
          ├── id (str)
          └── answers (dict)
              ├── answer_start (integer)
              └── text (str)
  ...
  └── paragraphs n
      ├── context
      └── qas
```

圖三：Json 格式的 SQuAD 資料集結構圖

首先，因為 SQuAD 資料集的格式複雜，但我們僅需要 SQuAD 資料集中的文章內容(context)及答案(answers-text)來做為訓練資料，因此我們必須從繁雜的資料集中，取出我們所需。並將句子跟答案做斷詞(tokenize)，並將取出的答案標記 answer，表示此段文字值得作為出題的方向。

由圖四我們看到，input 就是我們將輸入的文章內容(context)進行斷詞之後，再加上[CLS]開頭，及[SEP]結尾，為了讓 GPU 平行運算我們需要將 batch 裡的每個 input 都補上 zero padding 以保證它們長度一致。而最重要的即是把原始文本轉換成 PyTorch 中的 3 種 id tensors，token_tensor 是由 tokenizer 轉換成的索引值，segment_tensor 就是用來識別句子的界限，而 mask_tensor 則是用來界定 attention mask 的範圍(1 表示讓 BERT 關注該位置，0 則表示 Padding)。



圖四 ：BERT sentence pair encoding

3. 將 SQuAD 資料集的文本套用至 spaCy :

在(二)研究動機與研究問題中，雖然 spaCy 無法辨別較長的片段作為答題的關鍵字，但是卻在普通的人名、時間、地點都能正確抓出關鍵字，並且很有效率。因此，我們希望結合 SQuAD 資料集中答案較長的優點，並利用舊有的 SQuAD 的優點，來提升自動問句生成系統中，關鍵字的抓取。

(六) 參考文獻

[1] Ming Liu, Rafael A Calvo, and Vasile Rus. 2012. G-Asks: An intelligent automatic question generation system for academic writing support.
[2] Jack Mostow and Hyeju Jang. 2012. Generating diagnostic multiple choice comprehension cloze questions. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. Association for Computational Linguistics, 136–146.
[3] Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating Natural Questions About an Image. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 1802–1813.
[4] Pranav Rajpurkar, Robin Jia, Percy Liang. ACL 2018. Know What You Don't Know: Unanswerable Questions for SQuAD.
[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text.
[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2019.BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding(Minneapolis, Minnesota),Association for Computational Linguistics,4171–4186.
[7] https://github.com/huggingface/transformers
[8] Steven Bird, Edward Loper.2004. NLTK: The Natura Language Toolkit.ACL(Barcelona, Spain), Association for Computational Linguistics, 214–217.
[9] Honnibal, Matthew and Montani, Ines and Van Landeghem, Sofie and Boyd, Adriane.spaCy: Industrial-strength Natural Language Processing in Python.2020.
[10] Chan, Y. H., & Fan, Y. C. (2019, November). A recurrent BERT-based model for question generation. In Proceedings of the 2nd Workshop on Machine Reading for Question Answering (pp. 154-162).
[11] Zhao, Y., Ni, X., Ding, Y., & Ke, Q. (2018). Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 3901-3910).

(七) 需要指導教授指導內容

申請人穆冠蓁目前就讀中興大學資工三年級。在基本程式撰寫的能力部分，申請人已經具備獨立執行此大專學生研究計畫的能力。

針對本計畫需要進一步跟老師學習的內容初步列於下:

(1)BERT 模型的架構;

(2)學習文本生成的各種架構;

(3)了解各個資料集對訓練模型的優缺點；

(4)學習如何在 BERT 上建立自己的線性分類器;