科技部

110年度大專學生研究計畫申請書

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 一、綜合資料： | |  | |  | |  | |
|  | |  | |  | |
| 申 請 人 ︻ 學 生 ︼ | 姓 名 | | 穆冠蓁 | | 身分證  號　碼 | | B223415245 | |
| 就 讀 學 校、  科 系 及 年 級 | | 國立中興大學  資訊科學與工程學系三年級 | | 電　話 | | 0912169697 | |
| 學 生 研 究  計 畫 名 稱 | |  | | | | | |
| 研 究 期 間 | | 自110年7月1日至111年2月底止，計8個月 | | | | | |
| 計 畫 歸 屬 司 別 | | □自然司　　 　■工程司 　□生科司  □人文司(含科學教育領域) | | | | | |
| 研究學門代碼及名稱 | | [E41]智慧計算(資訊)[E4104]自然語言與語音處理 | | | | | |
| 上年度曾執行本部大專學生研究計畫 | | □是（計畫編號：M0ST － － － － ）  ■否 | | | | | |
| 指 導 教 授 | 姓 名 | | 范耀中 | | 身分證 號　碼 | | E122705705 | |
| 服 務 機 構 及 科 系(所) | | 國立中興大學資訊科學與工程學系(所) | | | | | |
| 職 稱 | | 副教授 | | 電　話 | | 0939003953 | |
| 補 助 經 費 | 每位學生每月6,000元研究助學金，研究期間為8個月，共計48,000元 | | | | | | | |

指導教授簽章： 申請人（學生）簽 章：

科、系主管姓名：

（學生就讀學校）

二、研究計畫內容

(一) 摘要

本研究欲透過「自動問句生成閱讀理解選項」為應用目標，欲透過提升問句生成過程中關鍵字篩選之難度及複雜度，來增進自動問句生成閱讀理解模型對未來教育環境的幫助。

(二) 研究動機與研究問題

近年來，因為自然語言處理領域的技術較成熟，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]。現今，市面上較為廣為人知且強大的NER (Name Entity Recogonition) API 有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。那麼比較好的結果應該是什麼？請說明於此 (add sentences to explain what will be a better result than “November 1922” and “January 1993”)

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。加一段預計如何透過BERT模型達成或者解決的說明於此。(add a paragraph here to explain how you plan to use SQuAD and BERT to achieve the goal you mentioned )

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)

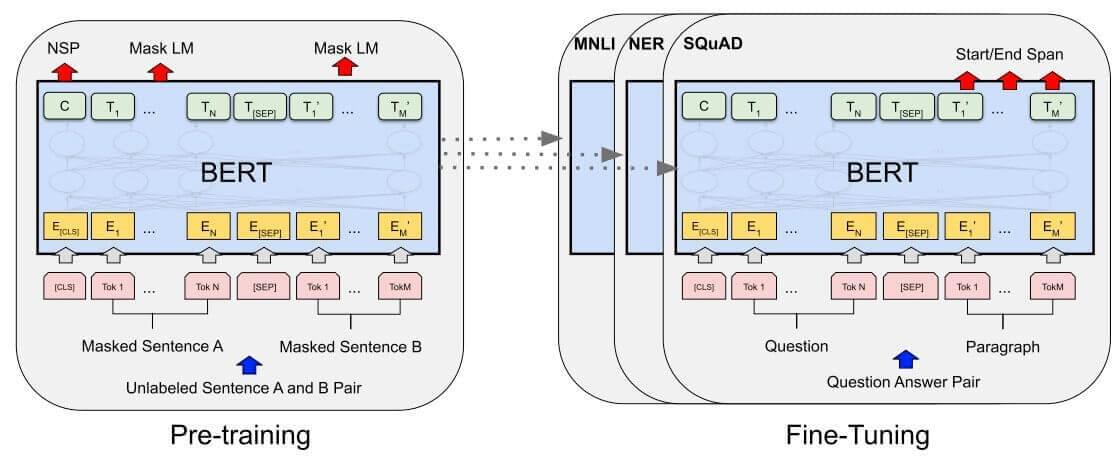
(三) 文獻回顧與探討

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]](#footnote-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。

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。



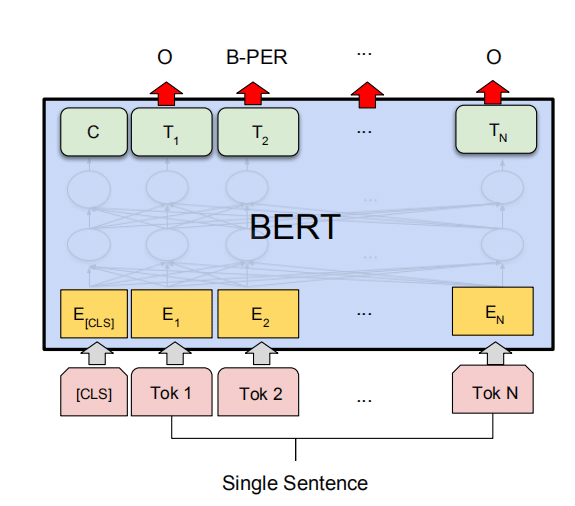
圖二 : Overall pre-training and fine-tuning procedures for BERT.

(四) 研究方法與步驟

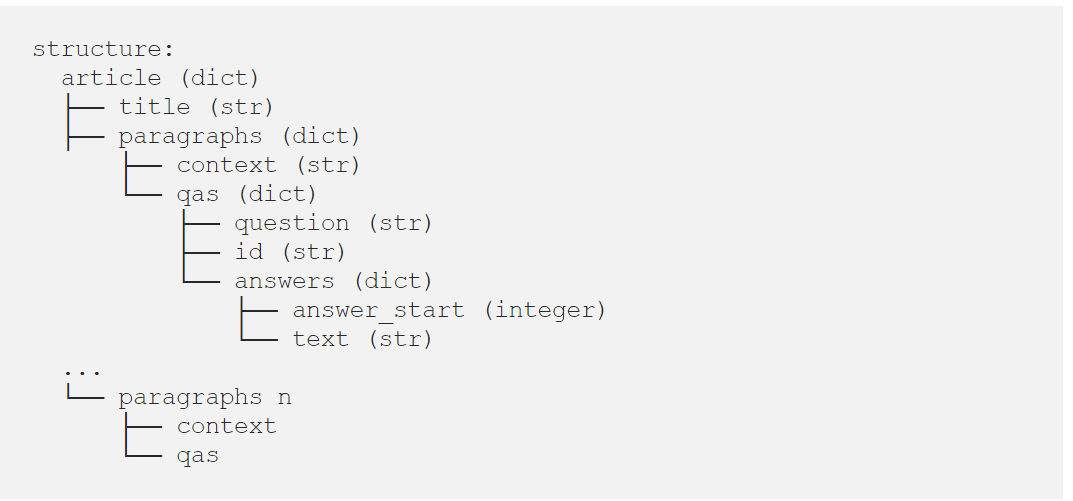
本計畫預計使用SQuAD作為訓練資料，並使用BERT 預訓練模型進行微調，來提升自動問句生成系統中的關鍵字選擇。而我們選擇SQuAD資料集的原因則是因為SQuAD是市面上第一個span-based(指問題的答案是從文章中標註出來的)的資料集，雖然span-based的特性同時限制了問題及答案的多樣性，但卻也使的問題的答案被限制在文章中。此外，SQuAD資料集克服了市面上現存的閱讀理解資料集的兩個問題(i)品質較高的資料集通常資料太小，無法訓練出有效的資料模型(ii)資料量大的資料集通常是半合成的，因此與平時所閱讀的閱讀理解題組不具有相同的特徵。也就是SQuAD資料集是現在資料量較大且兼顧品質的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

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。



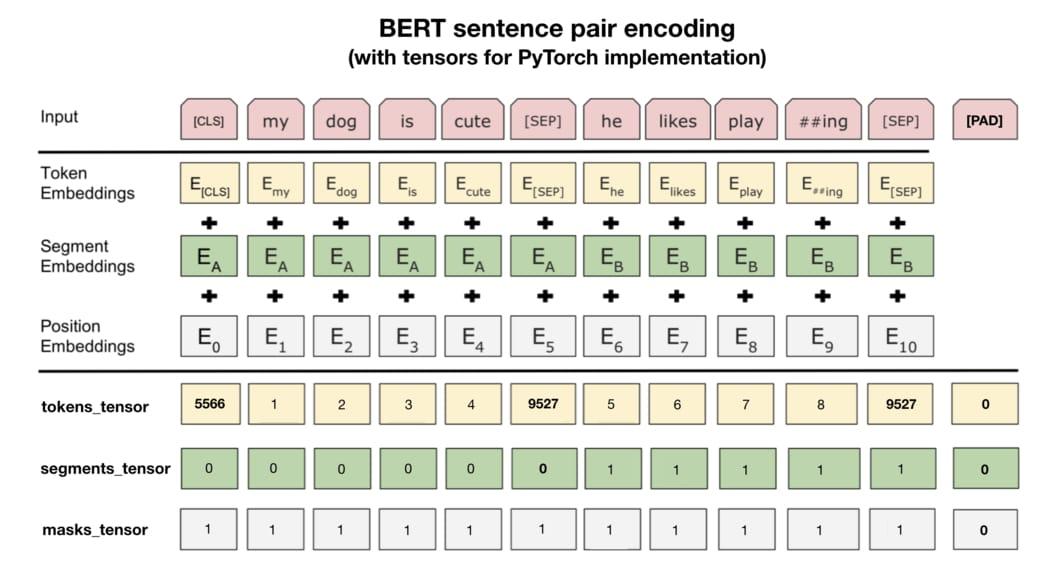
圖三: BertForTokenClassification

1. 將 SQuAD 資料集轉換成BERT相容的格式 :

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。

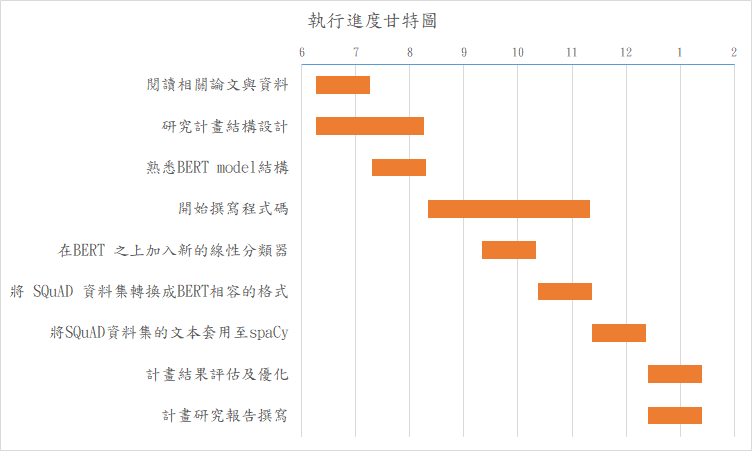
由圖四我們看到，input就是我們將輸入的文章內容(context)進行斷詞之後，再加上[CLS]開頭，及[SEP]結尾，為了讓 GPU 平行運算我們需要將 batch 裡的每個input都補上 zero padding 以保證它們長度一致。而最重要的即是把原始文本轉換成PyTorch中的3種id tensors，token\_tensor是由tokenizer轉換成的索引值，segment\_tensor就是用來識別句子的界限，而mask\_tensor則是用來界定attention mask的範圍(1表示讓BERT關注該位置，0則表示Padding)。



圖四 : BERT sentence pair encoding

1. 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。



(五) 預期結果

1. 設計出適合的Linear Classifier:

希望在了解BERT模型架構之後，能自行寫出我們想要用來分類這個token(字詞)是否適合作為答案的分類器。

2. 希望能讓自動問句生成系統更貼近人供出題的品質 :

欲透過整合SQuAD資料集、spaCy API的優點，使的挑選關鍵字的時候能更貼近老師的出題的方向。

1. 將此技術應用於開發英語學習系統 :

成功實現此技術後，希望能結合時下流行的通訊軟體LINE，我們預計在每天爬取CNN、Washington Post等國外新聞網站，並將文章爬取下來後，將本計畫的關鍵技術用於關鍵字的選擇，並在選擇完關鍵字之後，與自動問句出題API結合，完成閱讀測驗，並存在後端資料庫中。只要使用者閱讀完文章，就能回答跟這篇文章相關的閱讀測驗題，讓本專案的技術真正的應用於教育用途，也讓自然語言技術造福更多群眾。

(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.

DOI: <http://dx.doi.org/10.18653/v1/P16-1170>

[4] [Pranav Rajpurkar](https://arxiv.org/search/cs?searchtype=author&query=Rajpurkar%2C+P), [Robin Jia](https://arxiv.org/search/cs?searchtype=author&query=Jia%2C+R), [Percy Liang](https://arxiv.org/search/cs?searchtype=author&query=Liang%2C+P). ACL 2018.Know What You Don't Know: Unanswerable Questions for SQuAD

[5] [Pranav Rajpurkar](https://arxiv.org/search/cs?searchtype=author&query=Rajpurkar%2C+P), [Jian Zhang](https://arxiv.org/search/cs?searchtype=author&query=Zhang%2C+J), [Konstantin Lopyrev](https://arxiv.org/search/cs?searchtype=author&query=Lopyrev%2C+K), [Percy Liang](https://arxiv.org/search/cs?searchtype=author&query=Liang%2C+P). 2016.SQuAD: 100,000+ Questions for Machine Comprehension of Text

[6] [Jacob Devlin](https://arxiv.org/search/cs?searchtype=author&query=Devlin%2C+J), [Ming-Wei Chang](https://arxiv.org/search/cs?searchtype=author&query=Chang%2C+M), [Kenton Lee](https://arxiv.org/search/cs?searchtype=author&query=Lee%2C+K), [Kristina Toutanova](https://arxiv.org/search/cs?searchtype=author&query=Toutanova%2C+K). 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 Natural 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.Zenodo

(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)想出如何生成多個不一樣的distractor;

(4)如何探討各個方法的成效;

(5)科技論文撰寫與投稿;

(6)口語表達能力提升。

大專學生研究計畫指導教授初評意見表

一、學生潛力評估：

二、對學生所提研究計畫內容之評述：

三、指導方式：

四、本人同意指導學生瞭解並遵照學術倫理規範；本計畫無違反學術倫理。

指導教授簽名：

年 月 日

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. [↑](#footnote-ref-1)