一個基於 BERT 與孿生架構的檢索模型
A BERT-based Siamese-structured Retrieval Model

姜宏昀
Hung-Yun Chiang
國立臺灣科技大學
National Taiwan University
of Science and Technology
harrychiang0@gmail.com

陳冠宇
Kuan-Yu Chen
國立臺灣科技大學
National Taiwan University
of Science and Technology
kychen@mail.ntust.edu.tw

摘要

由於深度學習的發展，在以 Transformer 為架構的雙向編碼器 BERT 的帶領下，自然語言處理的相關任務獲得長足的進步。資訊檢索任務是從大量的文件中，尋找出與使用者查詢最相關的結果。雖然基於 BERT 的檢索模型已在許多研究中展現優異的任務成效，但這些模型通常有著計算量龐大或需要大量額外儲存空間的問題。有鑑於此，本研究提出一套基於 BERT 與孿生架構的檢索模型，不僅擁有以預訓練語言模型為主體的優點，更具備了自動查詢擴增技術，與使用強化學習於模型訓練。因此，我們所提出的檢索模型不僅改善了現有方法的問題，也在三個公開的大型資料集中，驗證了它的檢索成效。

Abstract

Due to the development of deep learning, the natural language processing tasks have made great progresses by leveraging the bidirectional encoder representations from Transformers (BERT). The goal of information retrieval is to search the most relevant results for the user's query from a large set of documents. Although BERT-based retrieval models have shown excellent results in many studies, these models usually suffer from the need for large amounts of computations and/or additional storage spaces. In view of the flaws, a BERT-based Siamese-structured retrieval model (BESS) is proposed in this paper. BESS not only inherits the merits of pre-trained language models, but also can generate extra information to compensate the original query automatically. Besides, the reinforcement learning strategy is introduced to make the model more robust. Accordingly, we evaluate BESS on three public-available corpora, and the experimental results demonstrate the efficiency of the proposed retrieval model.

關鍵字：BERT、資訊檢索、孿生架構、查詢擴增、強化學習

Keywords: BERT, Information Retrieval, Siamese-structured, Query Expansion, Reinforcement Learning

1 緒論

資訊檢索(Information Retrieval)是自然語言處理中一個重要的研究題目，目標是從大量的文件、段落或句子中，尋找出與使用者輸入之查詢(Query)最相關的答案。根據檢索內容的不同，資訊檢索任務又可分為文件檢索(Document Retrieval) (Yilmaz et al., 2019; Hofstätter et al., 2020; Mitra et al., 2020; Chen et al., 2020; Saar et al., 2020)與段落檢索(Passage Retrieval) (Cohen et al., 2018; Karpukhin et al., 2020; Khattab and Zaharia., 2020; Joel et al., 2020 ;Qu et al., 2021)。在過去的研究中，詞頻(Term Frequency) (Luhn, 1957)與反文件頻(Inverse Document Frequency) (Jones, 1972)是最常被使用的特徵表示法。詞頻是計算一個詞在文件中出現的次數，次數越高，通常代表這個詞在文件中是比較重要的；反文件頻則是一個詞出現在整個資料集中的文件比例之倒數，代表著這個詞的獨特性與鑑別性。藉由計算每一個詞的詞頻與反文件頻，文件與查詢可被分別表示為一組離散的特徵，藉由不同的檢索演算法，就可以計算每一篇文件

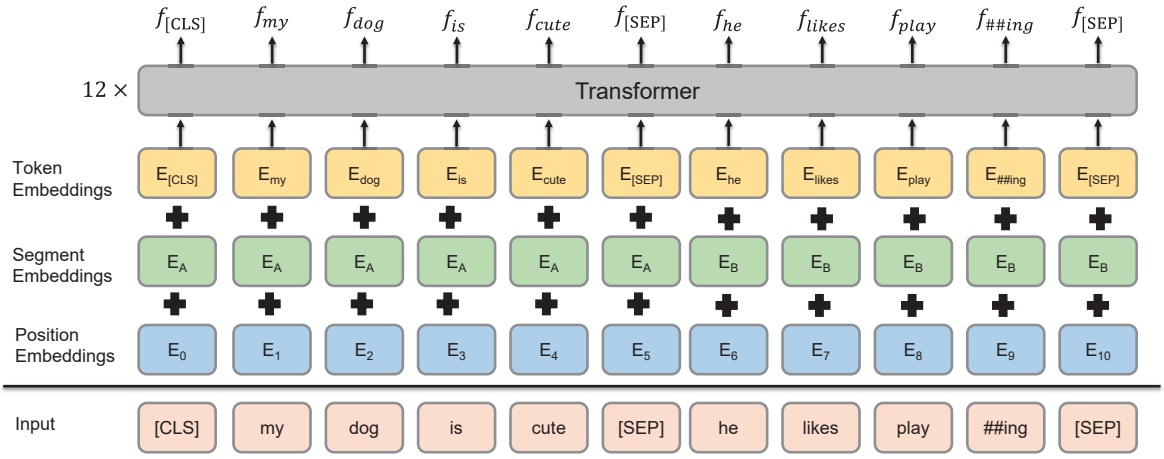
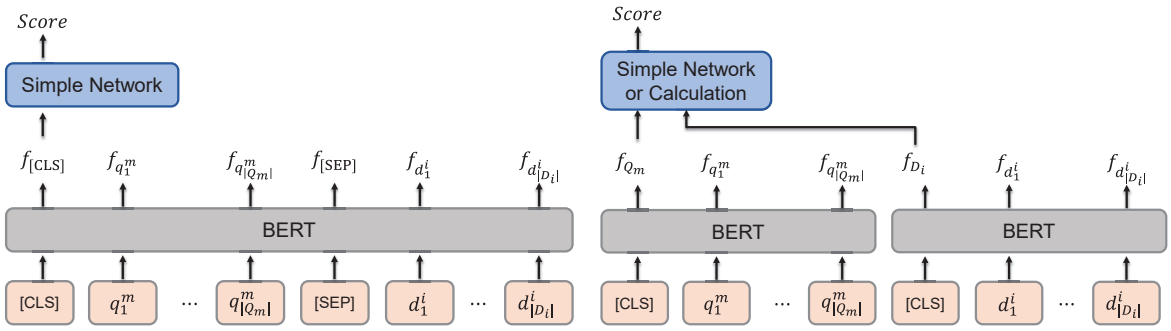


圖 1: BERT 模型架構圖。

與查詢的相關性分數，做為文件排序的依據並輸出。常見的檢索模型包含空間向量模型(Vector space model) (Salton et al., 1975)與 Okapi Best Match 25 (BM25) (Robertson et al., 1995)等。雖然這類方法簡單、快速，並且可以獲得相當的檢索成效，但僅透過關鍵詞匹配來計算相關性分數，不僅無法考慮查詢與文件的語意資訊，亦無法解決同義詞與一詞多義的問題。為此，後續有許多檢索模型紛紛提出，包含潛藏語意分析(Latent Semantic Analysis, LSA) (Deerwester et al., 1990)與主題模型(Topic Model) (Hofmann, 1999; Papadimitriou et al., 2000; Blei et al., 2003)等。

受惠於深度學習的蓬勃發展，自然語言處理的相關任務也在近期有了突破性的進展。以 Transformer (Vaswani et al., 2017)為主要架構的雙向編碼器 BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019)及各種變形模型，例如 XLNet (Yang et al., 2019)、RoBERTa (Liu et al., 2019) 與 Electra (Clark et al., 2020)等，皆是以非監督式的方式訓練一個語言模型，在預訓練(Pre-trained)的語言模型完成後，針對各式下游任務，這類模型僅需以少量的標記資料進行微調(Fine-tune)，就可以在該任務中獲得相當優良的任務成效。當使用 BERT 於資訊檢索時，最常見的作法是將查詢與文件串接後，藉由 BERT 抽取一個低維度的向量做為特徵，藉由簡單的前饋神經網路與軟性最大(Softmax)激活函數，計算出此一文件與查詢的相關程度。亦有方法是以孿生(Siamese)架構基礎，利用兩個 BERT 模型分別為查詢與文件進行特徵向量的抽取後，再利用餘弦相似度或藉由各種神經網路模型進行相關性分數的計算。著名的模型包含有 DPR (Karpukhin et al,. 2020)、SentenceBERT (Reimers and Gurevych., 2019)、TwinBERT (Lu et al., 2020)與 ColBERT (Khattab and Zaharia., 2020)等。

相較於傳統的資訊檢索模型，以 BERT 為基礎的模型可以獲得相當優良的任務成效，但這些模型通常有著計算量龐大或需要大量額外儲存空間的問題，雖然已經有些方法針對這些缺點加以改善，但其成果仍有待提升。有鑑於此，本研究提出一套基於 BERT 與孿生架構的檢索模型(**BE**RT-based **S**iamese-**s**tructured Retrieval Model, BESS)，不僅繼承著以預訓練語言模型為主體的優點，更著眼於改善現有模型時間與空間複雜度過高的問題。此外，考量使用者查詢通常較短，而容易產生資訊不足的問題，我們的模型設計了一套自動的查詢擴增(Query Expansion)技術；並且，在模型訓練的過程中，我們提出了一套權重計算方式，為每一個訓練查詢，根據當前的檢索結果，計算一個權重，做為更新檢索模型參數時的比重。綜合這些改進，我們在三個公開的大型資料集中，驗證了此一檢索模型的成效。實驗結果顯示，相較於各式基礎系統，這套新穎的檢索模型 BESS 不僅獲得相當良好的檢索成效，在測試階段亦擁有可接受的時間與空間複雜度。



(a) (b)

圖 2: (a) Cross-Encoder 模型架構圖。(b) TwinBERT 與 DPR 模型架構圖。

2 相關研究

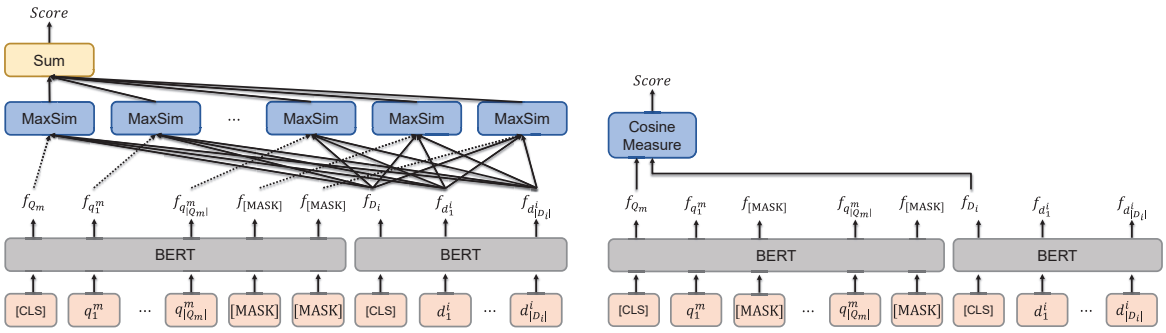
2.1 BERT & Cross-Encoder

基於 Transformer 的雙向編碼器 BERT (Bidirectional Encoder Representations from Transformers)，是一個以大量文本配合非監督式學習所訓練出來的語言模型，它已被廣泛使用在自然語言處理的各項任務中，並且皆能取得良好的任務成效。BERT 語言模型的訓練目標為克漏字任務(Masked Language Model)與語句關聯性預測(Next Sentence Prediction)。在克漏字的訓練中，會隨機屏蔽訓練語句中15%的字符(Token)，並用一個特殊字符[MASK]作為代替，希望模型可以根據上下文資訊，預測此一被替換掉的字符。語句關聯性預測則是將兩段語句串接後輸入 BERT，期望模型可以準確判斷這兩個語句是否為上下文的關係。為了考量字符的順序資訊，BERT 模型引入了位置向量(Position Embeddings)與段落向量(Segment Embeddings)。位置向量是用來表示每一個字符在語句中的絕對位置，而段落向量則用於表示字符是屬於第一個輸入語句或是第二個語句。最終，每一個輸入 BERT 的字符會表示成一個加總字符相量、位置向量與段落向量的向量表示法；此外，[CLS]與[SEP]為兩個特殊的字符，通常分別插入在每個輸入語句的最前面與兩個句子之間，圖 1 為 BERT 模型的示意圖。

當 BERT 被使用於資訊檢索任務時，最起初的作法是將查詢與文件當成兩個句子，串接在一起後輸入 BERT 模型，再利用最終的[CLS]向量作為融合查詢與文件的向量表示法，藉由微調一個簡單的分類器，輸出相關性分數，作為文件排序的依據，這類方法我們統稱為為 Cross-Encoder 模型(Rodrigo and Cho, 2019 ; Qu et al., 2021)，其架構如圖 2(a)所示。雖然 Cross-Encoder 能透過 BERT 很好地得到混合查詢與文件的向量表示法，進而計算相關性分數，但對於每一個使用者輸入的查詢，Cross-Encoder 必須將查詢與資料集內的所有文件一一串接，分別輸入 BERT 獲得混合兩者資訊的向量後，再計算分數。由於資料集中的文件數量通常非常多，因此 Cross-Encoder 是非常耗費時間的。

2.2 Siamese-structured Retrieval Models

因為查詢與文件的長度、內容複雜性與表達方式等性質有著不小的差異，有研究指出，查詢與文件的向量表示法應以不同的模型進行求取，因此以孿生架構為模型基礎的檢索模型應運而生。這類模型採用兩個獨立的 BERT 作為特徵抽取器，分別輸入查詢與文件的字符序列，而最後一層的[CLS]向量（或是對最後一層的字符向量進行加權平均），即被用來做為查詢與文件的向量表示法，透過簡單的餘弦相似度(Cosine Similarity)計算，或藉由簡單的神經網路架構，就可以獲得文件對於查詢的相關性分數。TwinBERT (Lu et al., 2020)與 DPR (Karpukhin et al,. 2020)是這類模型經典的代表，他們的模型架構如圖 2(b)所示。以孿生架構為模型基礎的好處是系統在實際應用時，所需面對的候選文件數量通常非常巨大，可能從數十萬篇至數千萬篇，但它們的內容通常是不會再改變的。由於這些特性，我們可以事先計算每一篇文件的向量表示法，並且將這些表示法儲存起來，當使用



(a) (b)

圖 3: (a) ColBERT 模型架構圖。(b) BESS 模型架構圖。

者輸入一個查詢後，我們僅需求取查詢的特徵向量表示法後，就可以跟已經算好並存儲起來的文件表示法進行相關性分數的計算，做為文件排序的依據。這樣的設計，在實際應用上，由於只需求取使用者輸入的查詢之特徵向量，並不需重複計算大量候選文件的向量表示法，因此可以大幅地減少所需的運算時間。

ColBERT (Khattab and Zaharia., 2020)同樣是以孿生架構為基礎的檢索模型，為了減少模型參數量，它的兩個 BERT 特徵抽取器的參數是共享的，為了區別查詢與文件的不同，在輸入時，將一個特別的字符[Q]加入在查詢的字符序列最前面，將特別字符[D]插入在每個文件字符序列的最前面。因此，雖然查詢與文件的特徵抽取器是參數共享的，但它依然可以區別輸入的是查詢或文件，而產生對應的向量表示法。此外，針對查詢通常遭遇資訊不足的問題，ColBERT 提出在查詢後面加入數個[MASK]字符，經過 BERT 後，這些[MASK]字符所對應的向量表示法，可以被視為是模型自動加入的查詢擴增資訊。最後，將查詢內的每一個字符向量與文件中每一個字符向量計算內積，再加總每一個查詢字符所得的最大分數，就可做為是查詢與文件的相關性分數，ColBERT 的模型架構如圖 3(a)所示。值得一提的是，雖然 ColBERT 可以預先將文件的向量表示法儲存起來，使得測試階段的速度可以較 Cross-Encoder 快，但相較於 TwinBERT 或 DPR，每篇文件僅儲存一個特徵向量，ColBERT 是將文件中所有字符的最後一層特徵向量皆儲存起來，因此 ColBERT 需要花費大量的記憶體空間。在計算相關性分數方面，由於 ColBERT 是將查詢中每一個字符向量與每一篇文件中的每個字符向量做內積計算，最後為每一個查詢中的字符留下一個最高的內積分數，加總後即為該篇文件對於查詢的相關性分數；然而，不論是 TwinBERT 或 DPR，文件的最後排序分數只要計算一個查詢特徵與一個文件特徵的內積，即是相關分數，因此 ColBERT 的計算複雜度也是 TwinBERT 或 DPR 的數千至數萬倍。

3 研究方法

3.1 模型架構

有鑑於基於 BERT 的模型已在資訊檢索任務中取得不錯的任務成效，此外，以孿生架構為基礎的模型可比 Cross-Encoder 有較佳的執行速度，因此，在本研究中，我們提出一套基於 BERT 與孿生架構之檢索模型(BERT-based Siamese-structured Retrieval Model, BESS)，模型架構如圖 3(b)所示。更明確地，當給定一個包含 M 筆資料的訓練集 $\boldsymbol{\Omega} = \langle Q_m, D_m^+, D_{m,1}^-, \dots, D_{m,N}^- \rangle_{m=1}^{M}$，每一筆資料包含一個長度為 $|Q_m|$ 個字符的查詢 $Q_m = [q_1^m, q_2^m, \dots, q_{|Q_m|}^m]$，一篇與 Q_m 相關的文件 D_m^+，其長度是 $|D_m^+|$ 個字符，以及 N 篇非相關文件 $\{D_{m,1}^-, \dots, D_{m,N}^-\}$。在基於 BERT 模型與孿生架構下，我們的模型擁有兩個參數不共享的 BERT 特徵抽取器，在查詢與文件的字符序列前後分別加上[CLS]與[SEP]後，即分別送入

特徵抽取器，並且以最後一層的[CLS]向量做為查詢或文件的特徵向量表示法$\langle f_{Q_m}, f_{D_m^+}, f_{D_{m,1}^-}, \cdots, f_{D_{m,N}^-} \rangle$。接著，模型的訓練目標函式為最大化負對數似然值(Negative Log-likelihood)：

$$\mathcal{L} = \sum_{m=1}^{M} -\log \frac{sim\left(f_{Q_m}, f_{D_m^+}\right)}{sim\left(f_{Q_m}, f_{D_m^+}\right) + \sum_{n=1}^{N} sim\left(f_{Q_m}, f_{D_{m,n}^-}\right)} \quad (1)$$

其中相似度函數$sim\left(f_Q, f_D\right)$定義為：

$$sim\left(f_Q, f_D\right) = exp\left(cos\left(f_{Q_m}, f_{D_m^+}\right)\right) \quad (2)$$

期望藉由錯誤傳遞，更新 BESS 模型的兩個特徵抽取器，使得查詢與相關文件可以擁有較相近的特徵向量，而非相關文件的特徵向量可以與查詢越不像越好。

3.2 查詢擴增

為了彌補使用者輸入的查詢通常較短，容易有資訊不足的問題，ColBERT 在查詢的字符序列後面加入多個[MASK]字符，讓檢索模型自動地為每一個查詢添加額外的資訊。在使用者給定的查詢中，名詞與形容詞往往是最為重要的資訊，並且借鑑於利用知識圖譜之 BERT 模型(Knowledge BERT, K-BERT) (Liu et al., 2020)的成功，我們延伸 K-BERT 模型的做法，期望能為查詢裡的名詞與形容詞添加一個自動產生的額外資訊。為了實現這個想法，我們替查詢裡的名詞與形容後面分別加入一個[MASK]字符，希望藉由訓練，BESS 可以自動地根據上下文資訊，補足名詞與形容詞資訊之不足，或是提供可能的額外資訊，使得檢索的成效可以更加提升。

3.3 強化學習

由於強化學習已在近年展現優異的成果(Arulkumaran et al., 2017; Yang et al., 2018; Satoshi and Toshihiko, 2020; Shao et al., 2021)，因此我們採用強化學習的方式訓練 BESS。為此，我們設計了一套訓練查詢權重函式，用來調整每一個訓練查詢對模型參數更新時的貢獻度，也就是扮演著強化學習中回饋(Reward)的角色，用來動態調整模型的學習率。至於如何判斷哪些訓練查詢對模型來說比較重要呢？我們首先採用一個簡易的檢索模型為訓練集中的每個查詢進行初次檢索，並計算檢索結果，例如準確率(Precision)或排序倒數平均值(Mean Reciprocal Rank, MRR)

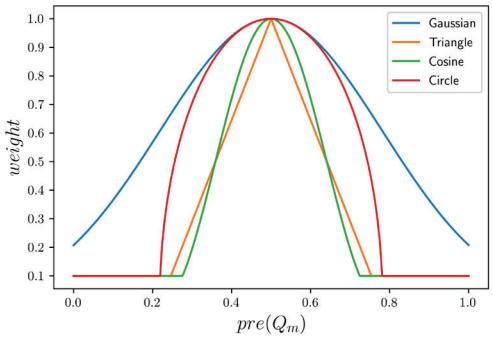


圖 4: 四種權重函式示意圖，以$\mu = 0.5$，$\sigma = 0.282$為範例。

(Hinrich et al., 2008)。有了每個訓練查詢的檢索結果後，我們提出四種不同的權重函式，包含高斯函式、三角函式、餘弦函式以及圓形函式(Lv and Zhai,. 2009)，用來計算每一個訓練查詢的權重：

- 高斯函式(Gaussian Kernel)

$$\frac{-(pre(Q_m)-\mu)^2}{2\sigma^2} \quad (3)$$

- 三角函式(Triangle Kernel)

$$\begin{cases} 1 - \frac{|pre(Q_m)-\mu|}{\sigma}, & if\ |pre(Q_m) - \mu| \le \sigma \\ 0.1, & otherwise \end{cases} \quad (4)$$

- 餘弦函式(Cosine Kernel)

$$\begin{cases} \frac{1}{2}\left[1 + cos\left(\frac{|pre(Q_m)-\mu|}{\sigma}\pi\right)\right], & if\ |pre(Q_m) - \mu| \le \sigma \\ 0.1, & otherwise \end{cases} \quad (5)$$

- 圓形函式(Circle Kernel)

$$\begin{cases} \sqrt{1 - \left(\frac{|pre(Q_m)-\mu|}{\sigma}\right)^2}, & if\ |pre(Q_m) - \mu| \le \sigma \\ 0.1, & otherwise \end{cases} \quad (6)$$

其中$pre(Q_m)$代表訓練查詢Q_m的檢索結果，μ與σ為權重函式的超參數，分別用來控制中心點與平滑程度，圖 4 為以μ =0.5 與σ =0.282 為範例的權重函式示意圖。與傳統的強化學習相較，傳統的回饋設計，通常是表現越差的訓練資料會給定較大的回饋，表現越好的資料會有較小的回饋；然而，我們認為，在使用簡易模型的檢索中，獲得良好成效的訓練查詢，不需要再對模型的訓練有較大的影響，因為這些查詢本身所蘊含的資訊與相關訊息，已被模型良好的描述與儲存，因此已可以擁有很好的檢索成果；另一方面，在簡

	Num. of Queries			Avg. Tokens/Query	Avg. Rel. Passages/Query	Num. of Passages	Avg. Tokens/Passage
	Train	Dev	Test				
MovieQA	38,417	4,333	4,327	8.80	1.55	86,360	99.39
MovieQA Chinese	38,417	4,333	4,327	12.85	1.37	107,340	167.88
MS MARCO	808,731	101,093	101,092	7.46	1.05	8,841,823	74.46

表 1: MovieQA、MovieQA Chinese 與 MS MARCO 資料集統計資訊。

易模型的檢索裡，獲得較差成效的訓練查詢也不需要對模型的更新有較大的影響性，因為這些訓練查詢可能含有錯誤的標記或屬於離群資料(Outlier)，給定較高的回饋，反而會影響模型整體的準確性。綜觀這些原因，我們希望表現尚可的訓練查詢才應對模型的更新有較大的影響，因此提出四種權重函式，並且，在本研究中，我們將μ設定為所有訓練查詢的準確率平均值。最終，模型的訓練目標函式則為：

$$\mathcal{L} = \sum_{m=1}^{M} -\log \frac{weight(Q_m) sim\left(f_{Q_m}, f_{D_m^+}\right)}{sim\left(f_{Q_m}, f_{D_m^+}\right) + \sum_{n=1}^{N} sim\left(f_{Q_m}, f_{D_{m,n}^-}\right)} \quad (7)$$

其中$weight(Q_m)$表示訓練查詢Q_m的回饋，可以由任一種權重函式計算獲得。

4 實驗與討論

4.1 資料集

本研究所使用的資料集包含MovieQA (Tapaswi et al,. 2016)、MovieQA Chinese (Tapaswi et al,. 2016)與 MS MARCO (Nguyen et al,. 2016)。MS MARCO 是一個由微軟在 2016 年推出用於閱讀理解任務的資料集，在 2018 年調整為段落檢索的資料集，包含了 880 萬個網頁中的段落，這些段落是 Bing 從 100 萬個實際的使用者查詢所收集而來，每個查詢皆對應到一個相關段落，但並沒有標註明確的非相關段落。在評量結果上，我們使用 MRR@10 與 MRR@100 來評估模型。所有實驗採用的訓練資料集是 MS MARCO small 版，並且基於官方所提供的初次檢索結果，對每一個查詢所對應的 1,000 個段落進行重新排序(Reranking)。因為該資料集的測試集沒有提供正確答案，因此在實驗中，我們隨機地切分訓練集的百分之十當做訓練時的驗證集，原始的驗證集則做為測試集使用。

MovieQA 是由多倫多大學所提供關於影片與文件的故事理解資料集，包含了 400 多部電影的相關文件，資料集中的每個問題都對應到一篇相關文件內的多個答案。我們將所有資料集中的文件切分成數個段落，每個段落大約包含 100 個單詞，因此每個查詢所對應的相關段落可能會有一至多個。為了驗證本研究所提出的方法是否可以應用在多種語言中，我們使用機器翻譯，將英文的 MovieQA 翻譯成中文，做為一套中文的資訊檢索資料集，資料前處理則與英文 MovieQA 資料集相同。與 MS MARCO 相較，在 MovieQA 與 MovieQA Chinese 資料集中，檢索模型是對所有文件進行排序，不是採用重計分的方式，僅對前幾篇文件進行重新排序。然而，因為 Cross-Encoder 所需的計算時間較長，因此我們先採用 BM25 進行初次檢索，再對前 1,000 則段落重新進行排序。在 MovieQA 與 MovieQA Chinese 資料集上，我們是以 MAP@10/50/100 進行模型的效能評估。資料集詳細的統計資訊如表 1 所示。

4.2 實驗設置

在英文的實驗中，我們使用 huggingface (Wolf et al., 2020)開源的 bert-base-uncased 模型，中文的實驗則使用 bert-base-chinese 模型。在我們所提出的檢索模型 BESS 中，查詢的字符序列長度設定為 32，文件字符序列長度設定為 384，若超過設定長度，則直接捨棄；模型訓練時的批次大小設定為 12；四種權重函式的超參數σ設定為 0.282，μ則根據資料集的不同，為 MovieQA、MovieQA Chinese 與 MS MARCO 分別設定為 0.725、0.64 與 0.364。我們的程式主要使用 pytorch 工具包，並利用 Faiss 工具包(Johnson et al., 2017)建立索引(Indexing)，與進行相關性分數的計算。

	MovieQA MAP@10/50/100	**MovieQA Chinese** MAP@10/50/100	**MS MARCO** MRR@10/100
BM25	38.4 / 39.1 / 39.2	33.3 / 34.1 / 34.2	16.7 / - (official)
Cross-Encoder	42.1 / 42.9 / 42.9	38.4 / 39.3 / 39.4	32.2 / 33.4
DPR	66.6 / 67.0 / 67.2	61.2 / 61.4 / 61.5	32.5 / 33.0
ColBERT	**70.4** / **70.9** / **71.0**	**63.5** / **63.9** / **64.0**	**33.4** / **34.3**

表 2: 基礎檢索模型於 MovieQA、MovieQA Chinese 與 MS MARCO 資料集之實驗結果。

4.3 實驗結果與討論

在第一組實驗中，我們首先探討基準系統在三個資料集的檢索成效，包含經典的 Okapi Best Match 25 (BM25) (Robertson et al., 1995)、基於 BERT 的 Cross-Encoder (Qu et al., 2021)還有屬於孿生網路架構的 DPR (Karpukhin et al,. 2020)與 ColBERT (Khattab and Zaharia., 2020)，實驗結果如表 2 所示。我們可以發現，以 BERT 為基礎的 Cross-Encoder、DPR 與 ColBERT 在三個資料集中皆大幅度的超越傳統 BM25 的成效，不僅說明預訓練語言模型所帶來的好處，也驗證了當前基於神經網路的檢索系統在大型資料集中的進步。接著，我們仔細比較 Cross-Encoder、DPR 與 ColBERT，基於孿生網路架構的 DPR 與 ColBERT，雖然需要花費額外的記憶體空間儲存文件的表示法，但他們不僅可以在測試階段擁有較快的運算速度，在檢索的成效上也可以獲得比 Cross-Encoder 要好的成績。最後，比較基於孿生網路架構的 DPR 與 ColBERT，因為 DPR 僅為每一篇文件儲存一個向量表示法，而 ColBERT 是將文件內所有的字符向量表示法皆儲存起來，由於實驗中，我們將文件的字符序列長度設定為 384，因此 ColBERT 所需的額外儲存空間大約是 DPR 的 384 倍；此外，ColBERT 在相關分數的計算上，是將每一個查詢的字符向量與每一個文件的字符向量進行內積計算，再整合出一個最終的分數，而 DPR 只需進行一次的內積計算，就可以獲得相關分數，因此 ColBERT 的計算複雜度幾乎是 DPR 的 12,288 (32×384)倍。雖然 ColBERT 的時間與空間複雜度皆比 DPR 高出許多，但實驗結果展現了 ColBERT 優異的檢索成效！

在第二組實驗中，我們測試本研究所提出之基於 BERT 與孿生架構的檢索模型 BESS 在三個資料集的檢索成效，實驗結果如表 3 所示。首先，$\text{BESS}_{\text{Gaussian}}$、$\text{BESS}_{\text{Triangle}}$、$\text{BESS}_{\text{Cosine}}$ 與 $\text{BESS}_{\text{Circle}}$ 分別表示使用四種不同權重函式的 BESS 模型，在 MovieQA 資料集中，使用高斯函式可以獲得最好的檢索成效，相較於餘弦函式，高斯函式甚至可以高出 4%的 MAP；在 MovieQA Chinese 資料集中，雖然圓形函式可以獲得最佳的檢索成效，但四種函式的效能差異不大；綜合比較 MovieQA 與 MovieQA Chinese 兩個資料集，使用餘弦函式的檢索成效皆是最差的，可能是因為餘弦函式給定的權重差異太大，即訓練查詢所獲得權重不是很大就是很小（參考圖 4），造成訓練時過分依賴部份資料而導致成效不彰的問題。接著，我們比較 BESS 與同為使用孿生架構的檢索模型 DPR 和 ColBERT。觀察表 2 與表 3，除了餘弦函式外，BESS 在三個資料集裡的檢索成效皆能大幅度的領先 DPR 模型，這個結果驗證了本研究所提出之自動查詢擴增與強化學習的有效性。與 ColBERT 相較，雖然 BESS 僅能獲得小幅度的成效提升，但值得一提的是，BESS 僅為每一篇文件儲存一個向量表示法，而 ColBERT 必須將文件內所有的字符向量表示法皆儲存起來，因為在實驗中，我們將文件的長度設定為 384，所以在額外儲存空間的花費上，ColBERT 的空間複雜度大約是 BESS 的 384 倍；在計算複雜度方面，因為 BESS 僅需為一組查詢與文件計算一次餘弦相似度，然而 ColBERT 是將查詢中的所有字符與文件中的所有字符兩兩計算餘弦相似度，再為每個查詢中的字符取最大值並相加，而實驗中，查詢的長度設定為 32，文件的字符序列長度設定為 384，因此 ColBERT 所需耗費的計算時間至少是 BESS 的 12,288 倍。綜觀上述，本研究所提出的 BESS 檢索模型，不僅在時間與空間複雜度上大幅度的優於 ColBERT 模型，在檢索任務的成效上，也可以取得與 ColBERT 相當或更佳的結果，基於孿生架構的設計，在

	MovieQA MAP@10/50/100	**MovieQA Chinese** MAP@10/50/100	**MS MARCO** MRR@10/100
$BESS_{Gaussian}$	**70.9** / **71.2** / **71.2**	63.4 / 63.6 / 63.6	**33.6** / **34.3**
$BESS_{Triangle}$	69.0 / 69.3 / 69.4	63.6 / 63.8 / 63.9	n/a
$BESS_{Cosine}$	66.1 / 66.7 / 66.7	62.5 / 62.7 / 62.7	n/a
$BESS_{Circle}$	70.0 / 70.2 / 70.2	**63.7** / **63.9** / **63.9**	n/a
BESS-RL	69.8 / 70.0 / 70.3	63.0 / 63.2 / 63.2	33.1 / 33.7
$BESS_{Gaussian}$-QE	70.1 / 70.3 / 70.4	62.8 / 63.0 / 63.0	32.7 / 33.3
$BESS_{Triangle}$-QE	68.8 / 69.1 / 69.2	63.1 / 63.3 / 63.3	32.4 / 33.1
$BESS_{Cosine}$-QE	65.7 / 66.0 / 66.1	62.2 / 62.4 / 62.4	32.3 / 32.6
$BESS_{Circle}$-QE	67.9 / 68.3 / 68.3	63.3 / 63.5 / 63.5	32.3 / 32.5

表 3: 本研究所提出之檢索模型 BESS 於 MovieQA、MovieQA Chinese 與 MS MARCO 資料集之實驗結果。

測試階段，BESS 也不需耗費大量的運算時間，藉由三個資料集，我們驗證了 BESS 的效率與能力！

在最後一組實驗裡，我們進行 BESS 模型的消融研究。當我們將強化學習取消，實驗結果如表 3 中 BESS-RL 所示，可以發現除了餘弦函式外，沒有使用強化學習的結果確實會讓大部分檢索的效能下降，這展現了權重函式為 BESS 的模型訓練帶來了一定的好處。此外，與 DPR 模型相較，這個實驗結果也說明了我們所提出的自動查詢擴增，在不同的資料集上，可以帶給檢索模型 1~3%的進步；接著，我們將自動查詢擴增取消，實驗結果如表 3 中的 $BESS_{Gaussian}$-QE、$BESS_{Triangle}$-QE、$BESS_{Cosine}$-QE 與 $BESS_{Circle}$-QE 所示，與 BESS 相較，缺少自動查詢擴增，不論在哪一種權重函式的使用下，皆會造成一定程度的效能損失。值得一提的是，雖然 ColBERT 與 BESS 皆有查詢擴增的設計，但由於在計算相關性分數時，ColBERT 是將查詢中所有字符的特徵向量皆與文件的每一個字符向量計算分數，因此 ColBERT 模型所擴增的查詢是直接的影響最後的排序分數，而 BESS 是只以特殊字符[CLS]向量做為查詢的特徵向量表示法，因此 BESS 模型的查詢擴增是以間接的方式改善最後的排序結果。從實驗中，我們可以說，自動查詢擴增，不論是以直接或間接的方式影響最後的排序結果，對於檢索任務的成效皆是有正向的幫助，但我們所提出的間接式方法，不僅可以提升檢索任務的成效，也不需要額外的計算負擔！

5 結論

在本研究中，我們提出了一套基於 BERT 與孿生架構的檢索模型 BESS，它不僅擁有良好的檢索效能，在測試階段，也不會有過高的計算負擔。此外，自動查詢擴增與強化學習的加入，更加提升了檢索模型的成效。我們在 MovieQA、MovieQA Chinese 與 MS MARCO 三個資料集中，驗證 BESS 模型的檢索能力，實驗結果顯示，BESS 不僅可以達到最好的檢索成果，也能有較低的計算複雜度。在未來的研究裡，我們將首先改進查詢擴增方法，使其更有效率；我們也將繼續驗證 BESS 模型於其他常見且公認的各式語言資料集中；除了資訊檢索外，我們希望能將 BESS 與開放式問答(Open Domain Question Answering)系統相結合，進一步地驗證，BESS 檢索模型是否能夠提升問答系統之成效。

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin. (2017). Attention is all you need. Advances in *Neural Information Processing Systems (p./pp. 5998--6008).*

Bhaskar Mitra, Sebastian Hofstatter, Hamed Zamani, & Nick Craswell. (2020). Conformer-Kernel with Query Term Independence for Document Retrieval. *arXiv preprint arXiv:2007.10434.*

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng & Jie Zhou. 2021. Sequence-Level Training for Non-Autoregressive Neural Machine Translation. arXiv preprint *arXiv:2106.08122*.

Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki & SantoshVempala (2000). Latent Semantic Indexing: A Probabilistic Analysis. *Journal of Computer and System Sciences, 61*(2), 217–235.

Daniel Cohen, Liu Yang, & W. Bruce Croft (2018). WikiPassageQA: A Benchmark Collection for Research on Non-Factoid Answer Passage Retrieval. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (pp. 1165–1168). Association for Computing Machinery.

David M. Blei, Andrew Y. Ng & Michael I. Jordan. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res., 3(null), 993–1022.*

Gerard M. Salton, Andrew Wong & Chungshu Yang (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11), 613–620.

Jacob Devlin, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171–4186). Association for Computational Linguistics.*

Jeff Johnson, Matthijs Douze & Herve J ´ egou. 2017. ´Billion-scale similarity search with GPUs. *ArXiv, abs/1702.08734.*

Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, & Anil Anthony Bharath. (2017). Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine, 34*(6), 26–38.

Kevin Clark, Minh-Thang Luong, Quoc V. Le & Christopher D. Manning. 2020. ELECTRA: Pretraining text encoders as discriminators rather than generators. *In International Conference on Learning Representations.*

Kosugi, Satoshi & Toshihiko Yamasaki. (2020, April). Unpaired image enhancement featuring reinforcement-learning-controlled image editing software. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 07, pp. 11296-11303).

Kuzi Saar, Zhang Mingyang, Li Cheng, Bendersky Michael & Najork Marc. (2020). Leveraging semantic and lexical matching to improve the recall of document retrieval systems: a hybrid approach. *arXiv preprint arXiv:2010.01195.*

Leslie Pack Kaelbling, Michael L. Littman, & Andrew W. Moore (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research, 4*, 237–285.

Luhn, Hans Peter (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development, 1*(4), 309-317.

Mackenzie Joel, Dai Zhuyun, Gallagher Luke & Callan Jamie. (2020, July). Efficiency implications of term weighting for passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 1821-1824).

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun & Sanja Fidler. (2016). *MovieQA: Understanding Stories in Movies through Question-Answering. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4631-4640.

Nils Reimers & Iryna Gurevych. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).*

Nogueira, Rodrigo & Kyunghyun Cho. (2019). Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085.*

Omar Khattab & Matei Zaharia. (2020). ColBERT. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Schütze Hinrich, Christopher D. Manning, & Prabhakar Raghavan. (2008). *Introduction to information retrieval* (Vol. 39, pp. 234-265). Cambridge: Cambridge University Press.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landaue & Richard Harshman. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell & Allan Hanbury. (2020). Local Self-Attention over Long Text for Efficient Document Retrieval. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.*

Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 60*, 493-502.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu & Mike Gatford. (1995). Okapi at TREC-3. *In Overview of the Third Text REtrieval Conference (TREC-3)* (pp. 109-126). Gaithersburg, MD: NIST.

Thomas Hofmann. (1999). Probabilistic Latent Semantic Analysis. *In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (pp. 289–296).* Morgan Kaufmann Publishers Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest & Alexander Rush. (2020). Transformers: State-of-the-Art Natural Language Processing. *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38–45). Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder & Li Deng. 2016. MS MARCO: A Human-Generated MAchine Reading COmprehension Dataset. (2016).

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen & Wen-tau Yih. (2020). Dense Passage Retrieval for Open-Domain Question Answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Wang, William Yang, Jiwei Li & Xiaodong He. (2018, July). Deep reinforcement learning for NLP. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts* (pp. 19-21).

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, & Ping Wang. (2020). K-BERT: Enabling Language Representation with Knowledge Graph. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(03), 2901-2908.

Wenhao Lu, Jian Jiao & Ruofei Zhang. (2020). TwinBERT: Distilling Knowledge to Twin-Structured BERT Models for Efficient Retrieval. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.*

Xuanang Chen, Ben He, Kai Hui, Le Sun & Yingfei Sun. (2021). Simplified TinyBERT: Knowledge Distillation for Document Retrieval. *arXiv preprint arXiv:2009.07531.*

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu & Haifeng Wang (2021). RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021* (pp. 5835–5847). Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer & Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692.*

Yuanhua Lv & ChengXiang Zhai. (2009). Positional language models for information retrieval. *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '09.*

Zeynep Akkalyoncu Yilmaz, Wei Yang & Haotian Zhang, Jimmy Lin (2019). Cross-Domain Modeling of Sentence-Level Evidence for Document Retrieval. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3490–3496). Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov & Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237.*