基於 Transformer 及遞歸神經網路的自然語言情感分析方法

(一)摘要

本計畫的目的是以遞歸神經網路(Recurrent Neural Network)實現 Transformer 注意力機制(Attention)並結合的過程。本研究來源以 BERT (Bidirectional Encoder Representation from Transformers) 、以 RNN 實現單頭注意力(Single Headed-Attention)及 Transformer 注意力機制為主軸進行研究。本研究計畫所使用的結合方法是基於 RNN 建構自然語言推理模型，並透過改造後的長短期記憶模型(LSTM)實現單頭注意力機制改良 Transformer 自我注意力機制弱於捕獲文本中的局部依賴問題，並取代 Transformer 中解碼器的注意力機制，進行自然語言情感分析任務。進一步以能進行平行處理的 Transformer 為基礎並融合改造後的 LSTM，改良現有的自然語言情感分析任務。希望本研究計畫所獲得的研究成果可以輔助並應用於社交網路分析、情感機器人及商品評價分析。

(二)研究動機與研究問題

◆ 研究動機

自然語言處理是結合人工智慧和語言學領域的重要方向，注重於自然語言與電腦之間的通訊互動。主要包含自然語言理解及自然語言生成，其中自然語言理解包含從自然語言的表達及語句中識別出該句真正的目的或含義，理解過程中會因段落句子組合不同而識別出不同的含義，也可能因一詞多義而理解錯誤。因此語義特徵的提取能力特別重要。自然語言生成部分，有能夠完成問答、閱讀理解、總結段落的能力。

以遞歸神經網路做為特徵提取，在提取過程中詞依先後順序讀入而被分配不同的權重，隨著詞與詞之間的距離拉遠及網路越深，越前面被輸入的權重會被稀釋，造成包含前面被輸入的資訊量會越來越少。以 Transformer 作為特徵提取可以避免先後輸入造成的權重稀釋問題，透過多頭注意力機制能夠於對當前預測詞，同時用到前面和後面的詞進行計算，對 Transformer 而言，能透過平行計算提升計算效率，相較之下 RNN 因依時序計算而效率不高。

由於 Transformer 與 RNN 主要差別在於解決時序輸入問題，但 Transformer 的無序輸入會造成段落句子組合不同而識別出不同的含義，因此加入位置編碼(Position Encoding) ，將位置編碼與詞嵌入向量相加作為輸入的嵌入向量。輸入過程中包含將輸入的嵌入向量透過多層的多頭注意力機制、前饋層 (Feed-Forward Neural Network)、和正規化(Layer Normalization) 。輸出過程包含以

殘差連接連接多層的多頭注意力和前饋層。

雖然 Transformer 的特徵提取技術在學界已相當成熟，但若當年 Transformer 並沒有成功發展並開源提供研究，而是以 LSTM 繼續做為自然語言處理的關鍵技術，在 Transformer 相當成熟的情況下，想像若以 LSTM 實作 Transformer 中關鍵的注意力機制效果如何? 因此，實作 SHA-RNN (Single-Headed Attention RNN)達到單頭注意力的技術受到矚目。

在 Transformer 的架構中用到多頭注意力(Multi-Head Attention)，雖然能避免順序輸入的缺失並提高預測效率，但卻無法確認每層的多個頭中有幾個是有效的，相較之下 SHA-RNN 的注意力只保留一個頭，訓練過程也能夠減少多餘的運算量。本研究希望藉由 LSTM 實作的 SHA-RNN 技術與 Transformer 中的編碼器進行合併以重組 Transformer 結構。

◆ **研究問題**

我們希望藉由 Transformer 及 SHA-RNN 重組編碼器和解碼器結構，能夠解決 Transformer 弱於捕獲短期文本依賴問題。Transformer 的運作過程因為會給予單一句子中出現的重複詞相同的權重，會造成無法給予相對鄰近詞有較大的權重，進而在自然語言理解也會出現問題。希望透過結合 SHA-RNN 技術能有助於解決 Transformer 弱於捕獲短期文本依賴問題，且能夠以單頭注意力提升運算效率。

本研究的主題如下，希望能夠過重組後的編碼器與解碼器進行更準確的自然語言處理。首先，對輸入語句以 BERT 進行預訓練，包含詞塊化 (tokenization) 及文本清理，進一步能夠判斷兩句話是否有相同的含義，並能夠捕捉句子之間的關係。透過 SHA-RNN 進行解碼，在注意力的過程中以 LSTM 實作單頭注意力以解決 Transformer 局部文本依賴問題，需要對向量進行矩陣乘法 (以 GeLU 做為激勵函數)，與傳統的下映射層相比能減少運算量。最後解碼器的部分將使用 Transformer 中經 SHA-RNN 改造後的解碼器用於自然語言生成。

因此，在本次研究中將對 Transformer 中的解碼器、SHA-RNN 注意力機制與多頭注意力機制之不同、Transformer 與 SHA-RNN 的融合效果進行分析和探討。

(三)文獻回顧與探討

一、Transformer

Transformer 實際上是一種基於自注意力機制的Seq2Seq(Sequence to Sequence)模型，以能夠提高神經網路機器翻譯性能的概念，不但能提高模型訓練的速度，也在特定任務中表現比 Google機器翻譯模型好，且相對於遞歸神經網路能透過平行計算以提升訓練速度。

✧ 自注意力機制 (Self-Attention)

Transformer編碼器的輸入首先會經過自注意力層，能夠幫助編碼器對特定單詞進行編碼的同時也查看輸入語句的其他單詞，之後自注意力層的

輸出會送到前饋神經網路。
解碼器也包含自注意力機制和前饋層，共兩層，但兩層之間是一層注意力層，能幫助解碼器將注意力集中在輸入語句的相關部分，如圖1。

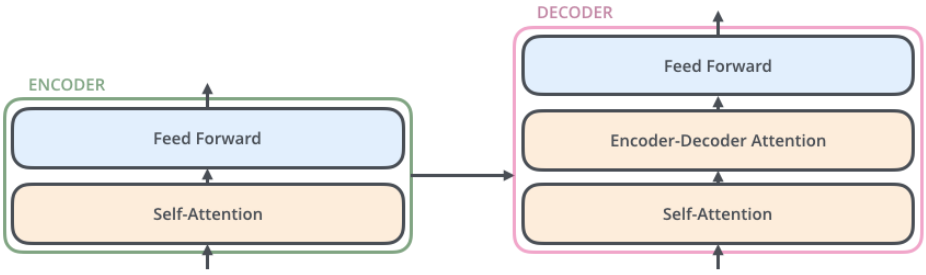


圖 1， 編碼器與解碼器架構圖， 來源: jalammer.github.io

對於輸入語句的每個單詞會建立三個向量（query查詢向量、key鍵向量、value值向量），透過將單詞的嵌入向量對訓練過程建立的三個矩陣(query, key, value）進行點乘（Dot-Production)產生向量。
得分將分別透過查詢向量的點乘與要得分的各個單詞的鍵向量相乘得出，為 q1·kn、q2·kn……再除以向量維度的平方根，以獲得更穩定的梯度。

最後透過softmax對分數進行標準化，softmax分數顯示該單詞對目標單詞的相關性。最後，將每個值向量乘以softmax分數，並加權總和就能得到自注意力輸出，如圖2。

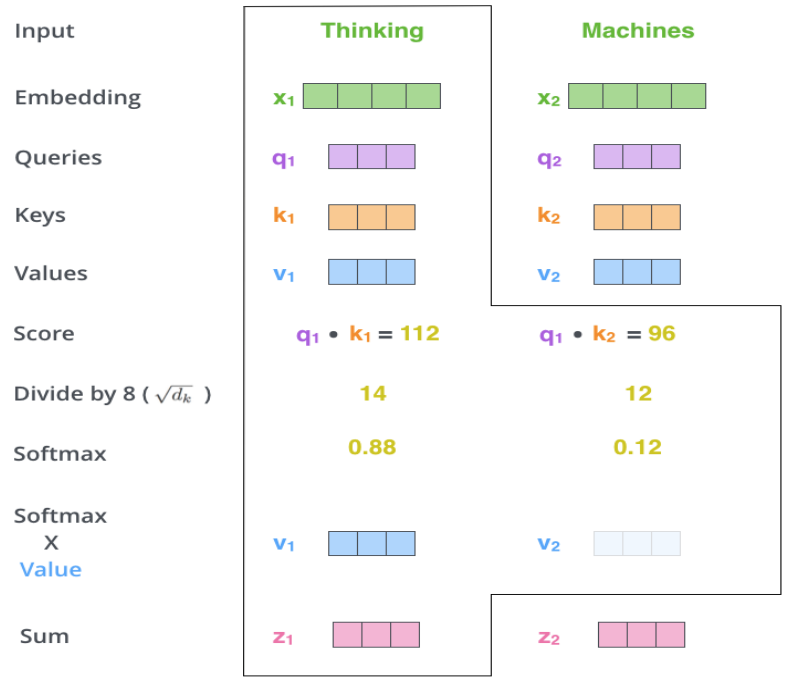


圖 2, 自注意力結構圖， 來源: jalammer.github.io

✧ 多頭注意力機制（Multi-Head Attention）

多頭注意力為單頭注意力的原理延伸，多頭注意力著重於單一查詢矩陣能夠和多個鍵向量進行點乘並能夠一起考慮整個輸入語句的單詞，如圖3。

此外，根據不同下游任務不同的頭關注的點不一樣，其中包含單頭取局部的資訊或以多頭取全局的資訊。因此，無論是單頭注意力機制或多頭注意力機制，在特定問題的解決方案都可能會用到，如圖4。

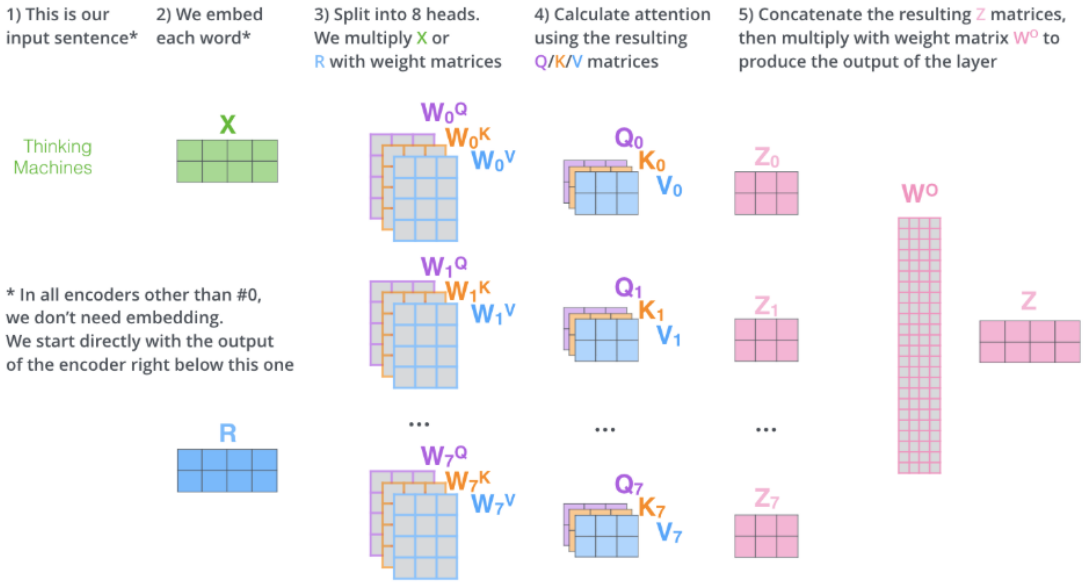


圖 3, 多頭注意力結構圖, 來源: jalammer.github.io

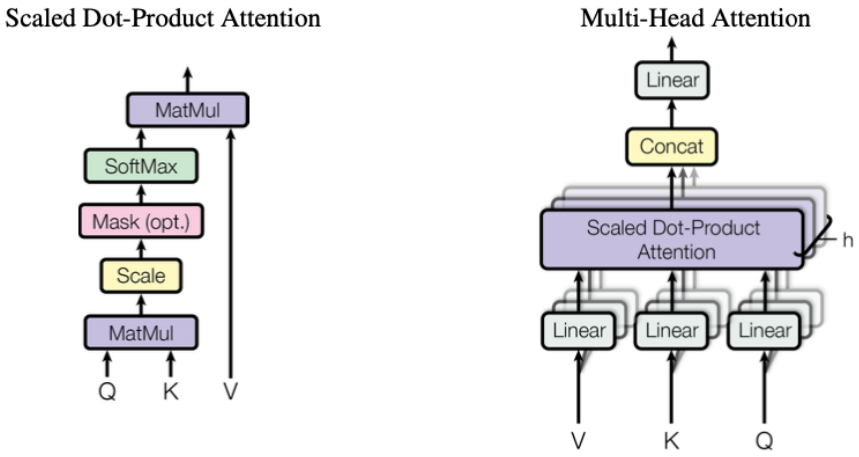


圖4, 注意力機制比較圖, 來源: Attention Is All You Need 原始論文

編碼的過程中，與傳統RNN相比確實能夠降低順序輸入的問題，但也因為無序問題而衍生出位置編碼（Position Encoding）的必要性。在未加入位置編碼的情況下，自注意力機制會將句子中出現重複的詞賦予一樣的權重，會造成相近的詞應較重要，但權重卻與另一個較遠的詞一樣重要，產生 Transformer 弱於捕獲文本中的短期依賴問題，這種對注意力的依賴可能會導致 Transformer在語法敏感任務上的性能不如RNN模型。

相較之下，RNN模型是過度仰賴短期依賴，但 Transformer 卻是缺少必要的短期依賴。

二、Bidirectional Encoder Representations from Transformers (BERT)

✧ 預訓練（Pre-Training）雙向 Transformer

傳統的語言模型因由數學定義為單向而且LSTM只能完成淺層訓練，導致對於不同位置方向的單詞而言，在編碼的過程看不到另一側的單詞。雖然句子中有些單詞會依賴鄰近左右側的單詞，但僅僅從單方向做編碼無法滿足需求。

透過 Transformer，能將網路做得更深，每個位置的詞都能不因位置距離和方向而進行編碼，但在自然語言生成的組合部分，雖然 BERT做詞嵌入時有加入位置編碼（Position Encoding），但其原理是被用來與輸入嵌入求平均來進一步能夠感覺詞塊的相對位置，因此認為語言組合可能也牽涉到詞序推理，並不是只需要注意力機制（Attention isn't all you need.）。

✧ 預訓練任務 #1 Masked Language Model（Masked LM）

透過訓練一個語言模型，大量未標註的數據進行無監督學習，語言模型能夠學會語法結構、解讀語義，並透過 BERT 能減少不同自然語言處理任務的預訓練和建構成本。

在訓練過程中會對訓練集隨機遮蔽 15%的單詞，而不是如以往將每個詞都預測一次。而最後損失函數也只會計算被遮蔽的詞塊（token），被遮蔽的15%其中會有10%被替換成其他單詞，而另外10%不替換，剩下80%才被替換為 [MASK] 。

在預訓練的過程中，模型不知道真正哪些詞被遮蔽，因此模型對每個詞都會注意，這也造成當[MASK]出現過多將影響模型收斂速度，甚至會比RNN左到右的模型更慢。

✧ 預訓練任務 #２ Next Sentence Prediction（NSP）

判斷第二個語句在原始文本中是否與第一個句子相接。

✧ 現今 NLP 的兩階段遷移學習

先以語言模型預訓練的方式訓練出一個對自然語言有相當程度理解的語言模型，再將模型用來做特徵擷取或針對下游任務進行微調。而透過BERT能夠同時完成無監督學習和監督式微調的部分。

三、Sequence to Sequence (Seq2Seq)

✧ 編碼器與解碼器

Seq2Seq模型主要由編碼器與解碼器兩個RNN組成，編碼器負責將輸入序列編碼轉換成中間向量(Context Vector)，解碼器再根據中間向量轉換成文字輸出。在預測的過程中，目前字詞的預測不僅取決於前面已翻譯的字詞，還必須考慮到原始輸入。

運作過程中，編碼器最後時間神經元的隱藏層會輸出到解碼器的第一個神經元，透過激勵函數和softmax層，篩選出機率最大者做為下一個神經元的輸入，如圖4。

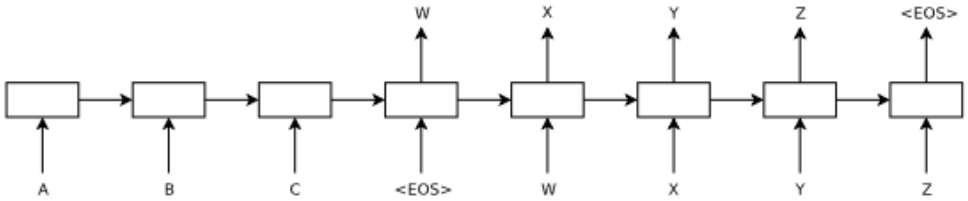


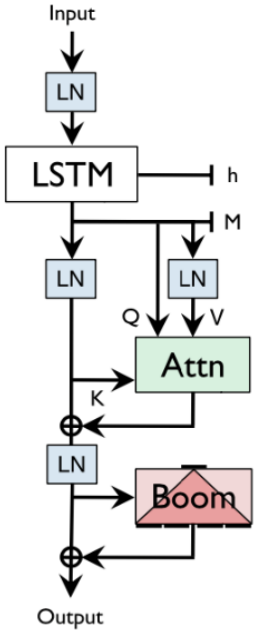
圖 4, Seq2Seq 結構圖, 來源: Seq2Seq 原始論文

而問題會出現在中間向量，在編碼器以最後一個神經元進行轉換時，雖然都是依序由左到右讀取資訊，但中間向量卻是固定長度維度的向量。導致轉換後的向量無法涵蓋到所有輸入序列的訊息，造成先被輸入的重要訊息可能在轉換後權重會變低或甚至消失。

四、Single Headed Attention RNN (SHA-RNN)

✧ 單頭注意力(Single Headed Attention)

Transformer模型是建立在無序基礎，並只透過注意力機制完成訓練，但每層網路都有幾十個頭(Head)，運算的過程中將因無法得知哪些頭為有效而耗費多餘的運算資源。
相較之下，SHA-RNN的注意力機制只保留一個頭以完成向量的注意力點乘。



✧ 結構變更

主要以Transformer的自注意力機制為基礎修改，四層的結構中，每層都是先做LSTM，進行層標準化(Layer Normalization)後再接上注意力機制，因此實際上為8層。

結構變更的部分，與Transformer相比之下只對Q做全連接層，並以sigmoid產生Q, K。但實作過程發現LSTM的輸出必須經過全連接層才成為Q,K,V，而改進的核心部分在於作者提出改造後的前饋層(Boom Layer)。

✧ Boom Layer

為了減少運算量，基於Transformer前饋層改造後的Boom Layer，將原本的全連接層轉換成4倍向量，加總後再將維度轉換回來，透過實作能夠節省顯存用量，跑更多層網路。

(四)研究方法及步驟

◆ **以 BERT 對原始文本進行預訓練 (Pre-training)及針對不同的下游任務進行微調 (Fine-tuning)**

BERT的運作過程主要基於未標註或只有少量標註的文本數據進行微調以解決新的下游任務。運作過程包括三個主要步驟:

1. 準備原始文本數據:

文本數據包含未標註或少量標註的文本，透過數據清理將文本中空白標題的範例去除。同時，將超出BERT模型中預設序列長度的範本去除，並以0將將小於序列長度的向量補0，以符合預訓練的文本讀入。

2. 將原始文本轉換成BERT相容的輸入格式:

文本進行預處理過程中會對句子開頭向量位置加入分類符[CLS]，並以[SEP]以0/1區分第一句與第二句。再以中文BERT對文本進行斷詞，如圖5。

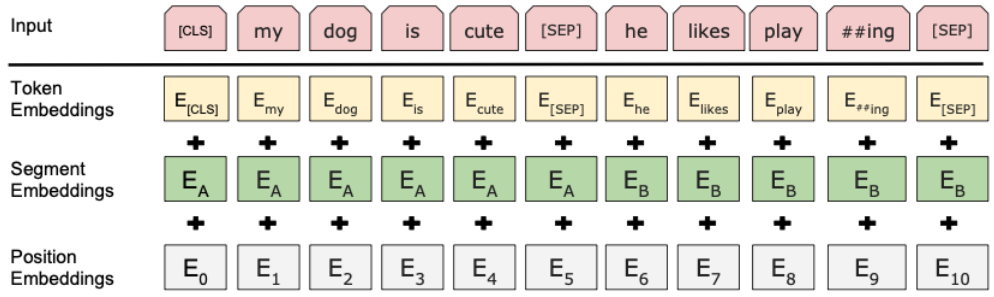


圖 5, BERT 成對句子編碼示意圖, 來源: BERT 原始論文

3. 在BERT之上加入新的Layer進行微調成為下游任務模型:

對BERT模型進行微調的部分包含利用下游任務的目標函式從頭訓練分類器並微調BERT參數，以訓練完的BERT加上線性分類器最大化當前下游任務的目標，如圖6。

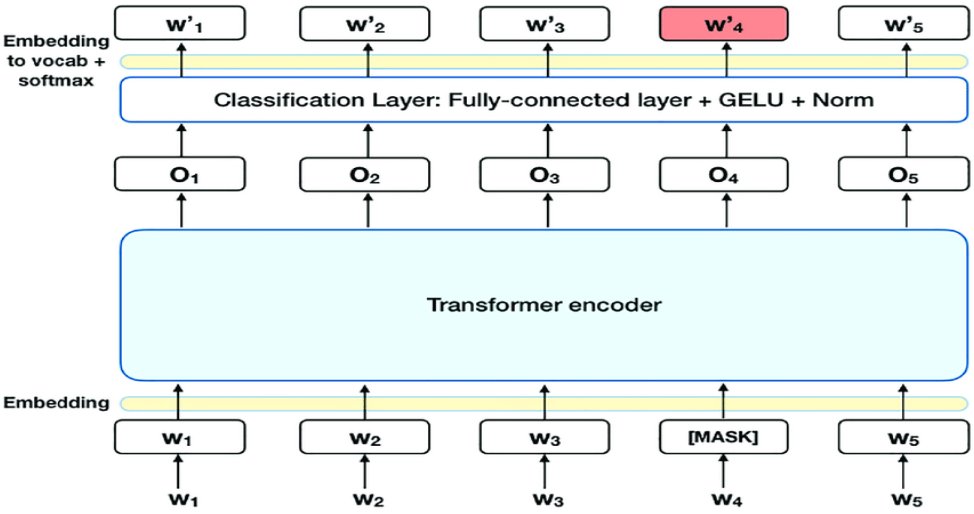


圖 6, BERT分類層示意圖, 來源: Faiza Khattak

透過遷移學習，新增的分類器由於大多數的參數都來自已經預訓練的BERT，實際上需要從頭訓練的參數量很少。

因此，在微調的過程中，需依照不同下游任務加入新的線性分類器。

◆ **以 SHA-RNN 實作注意力機制**

將 BERT 針對不同下游任務進行微調後的模型依照不同注意力產生預測的字詞放入成為 SHA-RNN 的輸入向量。

SHA-RNN 運作過程包含解碼過程的注意力分配和前饋層的向量運算，其中主要包含兩個核心結構，基於指針的注意力(Pointer Based Attention)和作者改造過的前饋層(Boom Layer)。

1. 基於指針的注意力機制 (Pointer Based Attention):

在運算過程中只保留一個頭，其中唯一的矩陣乘法只出現於 Query，經過矩陣相乘及層級標準化,如式(1)及式(2)，其中 h^{t-1} 為 t-1 時刻的隱藏層狀態，和 t 時刻的輸入數據 x_t。

再進行縮放點乘注意力(Scaled Dot-Product Attention)，如式(3)。

$$\mathbf{a}^t = W_{hh} h^{t-1} + W_{xh} \mathbf{x}^t \tag{1}$$

$$\mathbf{h}^t = f\left(\frac{\mathbf{g}}{\sqrt{(\sigma^t)^2 + \epsilon}} \odot (\mathbf{a}^t - \mu^t) + \mathbf{b}\right) \qquad \mu^t = \frac{1}{H}\sum_{i=1}^{H} a_i^t \qquad \sigma^t = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(a_i^t - \mu^t)^2} \tag{2}$$

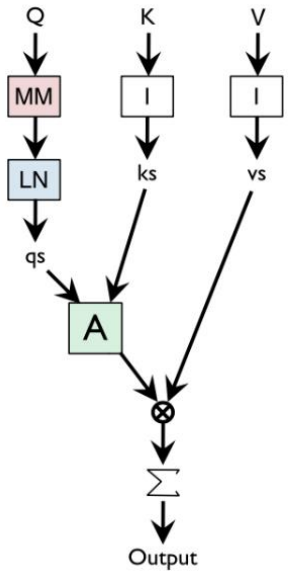
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

◆ 改造後的前饋層(Boom Layer):

類似於 Transformer 中改造的前饋層，使用一個 $v \in \mathbb{R}^H$ 向量，透過以激活函數 GeLU 的矩陣乘法，得到 $u \in \mathbb{R}^{N \times H}$ 之後將 u 拆成 N 個向量，再加總得到 $w \in \mathbb{R}^H$。

$$v \in \mathbb{R}^H \longrightarrow u \in \mathbb{R}^{N \times H} \longrightarrow w \in \mathbb{R}^H$$

圖 7, 向量維度變化示意圖



◆ **解碼輸出**

透過將 Transformer 中的編碼器部分以 BERT 進行預訓練和監督式微調，並將 BERT 輸出向量做為 SHA-RNN 的輸入向量。再透過 SHA-RNN 進行解碼輸出，即完成本研究所有的研究步驟。

下圖左側為原始 Transformer 架構圖，右側為本研究提出改造後的新架構，如圖 8。

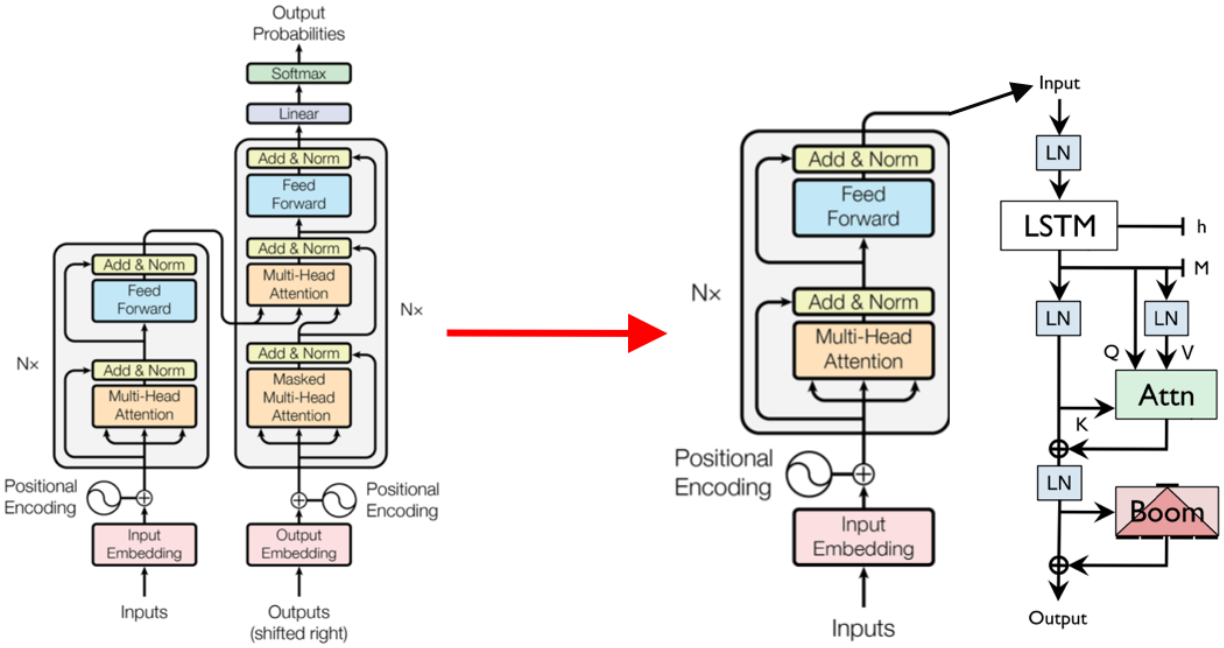


圖 8, 研究架構圖

(五)預期結果

經由此研究步驟，本計畫預期可獲得由原始文本經由 BERT 進行預訓練及監督式微調的結果，透過 BERT 編碼器的輸出並以 SHA-RNN 以單頭注意力完成解碼器的前饋層及注意力機制，最後得出比原始 Transformer 詞義分析輸出更精準的預測結果，並能夠彌補 Transformer 注意力機制導致編解碼過程無序的缺失以及其衍生出的弱於捕捉文本依賴問題。最後將應用於社交網路分析、情感機器人及商品評價分析或後續其他自然語言情感分析任務。

(六)參考文獻

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv: 1810.04805v2, 2019.
2. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jacob Uszkoreit, Llion Jones,

Aidan N.Gomez, Lukasz Kaiser and Illia Polosukhin. Attention Is All You Need. arXiv: 1706.03762v5, 2017.
3. Yoav Goldberg. Assessing BERT's Syntactic Abilities. arXiv: 1901.05287v1, 2019.
4. Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, 2013.
5. Stephen Merity. Single Headed Attention RNN: Stop Thinking With Your Head. arXiv: 1911.11423v2, 2019.
6. Ilya Sutskever, Orial Vinyals and Quo V.Le. Sequence to Sequence Learning with Neural Networks. arXiv: 1409.3215v3, 2014.
7. Nikita Kitaev, Lukasz Kaiser and Anselm Levskaya. Reformer: The Efficient Transformer. arXiv: 2001.04451v2, 2020.
8. Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V.Le and Mohammad Norozui. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. arXiv: 1609.08144v2, 2016.
9. Stephen Merity, Caiming Xiong, James Bradbury and Richard Socher. Pointer Sentinel Mixture Models. arXiv: 1609.07843v1, 2016.
10. Lajanugen Logeswaran and Honglak Lee. An Efficient Framework for Learning Sentence Representation. arXiv: 1803.02893v1, 2018.

(七)需要指導教授指導內容

一、請老師指導文獻蒐集的要領。
二、請老師指導論文研讀與整理的要領。
三、請老師指導如何進行實驗設計。
四、請老師指導如何提供有效的特徵，以及擷取特徵的技巧。
五、請老師指導撰寫程式的技巧。
六、請老師指導研究報告撰寫的寫作要領。