

## 應用多跳躍注意記憶關聯於記憶網路之研究

# A Research of Applying Multi-hop Attention and Memory Relations on Memory Networks

詹京翰\*、劉立頌\*、李俊宏†

Jing-Han Zhan, Alan Liu, and Chiung-Hon Lee

### 摘要

機器學習與深度學習近年發展越來越迅速，在自然語言處理任務上取得相當大的突破。透過類神經網路可以實現複雜的語言任務，如文章分類、摘要提取、問答任務、機器翻譯、圖片說明生成等。本論文以記憶網路做為研究目標、問答任務作為驗證應用。模型將先驗知識保存於記憶中，再透過注意力機制 (Attention Mechanism) 找出與問題相關的記憶，並推理出最終答案。問答任務數據集採用 Facebook 所提供的 bAbI 數據集，其中共有 20 項不同種類的問答任務，可驗證模型在不同任務的準確率。此研究透過記憶間的關聯計算，縮減記憶關聯的數量，除了下降 26.8% 權重的計算量外，也能提高模型的準確率，於實驗中最多可提高約 9.2% 左右。同時實驗採取較小的數據量作為驗證目標，改善即使在數據集不足的情況也能達到相當程度的改善效果。

### Abstract

With the rapid advancement of machine learning and deep learning, a great breakthrough has been achieved in many areas of natural language processing in recent years. Complex language tasks, such as article classification, abstract extraction, question answering, machine translation, and image description generation, have been solved by neural networks. In this paper, we propose a new

---

\*國立中正大學電機工程學系

Department of Electrical Engineering, National Chung Cheng University

†南華大學資工系

Department of Computer Science and Information Engineering, Nanhua University

E-mail: hoe8624@gmail.com; aliu@ee.ccu.edu.tw; chlee@nhu.edu.tw

The author for corresponsence is Chiung-Hon Lee.

model based on memory networks to include a multi-hop mechanism to process a set of sentences in small quantity, and the question-answering task is used as the verification application. The model saves the knowledge in memory first and then finds the relevant memory through the attention mechanism, and the output module reasons the final answer. All experiments have used the bAbI dataset provided by Facebook. There are 20 different kinds of Q&A tasks in the data set that can be used to evaluate the model in different aspects. This approach reduces the number of memory associations through the calculation of associations between memories. In addition to reducing the calculation weight of 26.8%, it can also improve the accuracy of the model, which can increase by about 9.2% in the experiment. The experiments also used a smaller amount of data to verify the system for improving the case of insufficient data set.

**關鍵詞：**記憶網路、多點跳躍網路、關係網路、注意力機制

**Keywords:**Memory Networks, Multi-hop Networks, Relation Networks, Attention Mechanism

## 1. 緒論 (Introduction)

深度學習研究近年大幅成長，其中記憶網路模型於文字相關的自然語言任務受到了不少關注。在自然語言領域中，聊天機器人、問答任務等，都具有序列資料的特性，也就是文句字詞有時間先後的關係，因此計算的過程需要給與模型詞序資訊，或是依照順序輸入至模型內。其中記憶模型使用外部記憶的方式儲存文本或先驗知識，推理時再從中找出與問題關聯性高的記憶內容，可以避免在計算過程中損失重要的資訊。其中結合了注意力機制的應用，使輸出模組可以根據現在的問題關注重要的記憶內容，推理得出正確的答案。

應用於語言模型領域中的記憶網路模型有單跳躍注意機制(Single-hop attention) 與多跳躍注意機制(Multi-hop attention) 的推理方式，因記憶與記憶間相互獨立，在數據量足夠大的狀況下已可學習到不錯的效果。但在數據量較小的前提下則較為無法學習數據內的資訊。為提高模型學習與推理的能力，提高記憶儲存效率與模型的推理方法則顯得相當重要。

本研究結合自然語言常用資料集與深度學習的工具，嘗試結合不同理論，強化語言理解與推理能力，並分析實驗結果的表現。實驗採取較小的數據量作為驗證目標，改善即使在數據集不足的情況也能達到相當程度的改善效果。目的歸納如下兩點。

- (一) 研究多跳躍注意機制對於記憶網路預測的表現。
- (二) 研究記憶網路記憶關聯提取對於推理能力的提升。

本論文研究在小數據集的前提下，不同的機制對於問答系統模型的影響。在研究中我們將關係網路的概念，以關聯記憶的形式與記憶模型結合，於 bAbI 數據集(Weston,

Bordes, Chopra & Mikolov, 2016)20 項任務中驗證，準確率最多可提高約 9.2%左右的準確率。關聯提取的部分還有降低權重的功用，相比於保存所有關聯計算，平均每項任務可下降 3 萬個權重數量，整體下降 26.8%權重的計算量。

我們在接下來的小節討論與整理記憶網路相關領域研究文獻；第三節為研究方法與設計，對於本論文研究方式與方法做一系列的整理與說明；第四節則為實驗結果與分析，比較改善前後模型的表現，驗證所採用方法的可行性與價值；最後一節為結論與建議，總結本論文所採用方法的優缺點，以及未來可嘗試的方向。

## 2. 文獻回顧 (Literature Review)

記憶網路(Memory Networks)主要運用於問答任務與情感分析等應用上，採用外部記憶的方式儲存先驗知識，透過注意力機制找到與問題相關的記憶內容，再利用推理模組從問題與相關記憶得出最終答案。記憶網路由許多模組組合而成，各個部分可由設計者採用不同方式實現，本小節介紹記憶網路相關研究，以及語言模型的相關理論。

### 2.1 注意力機制 (Attention Mechanism)

注意力機制(Attention mechanism)最早應用於圖像領域，論文(Bahdanau, Cho & Bengio, 2015)結合類神經網路模型，將其運用於機器翻譯任務上，首次將注意力機制應用於自然語言處理領域上。

自從編碼器解碼器架構(Encoder-Decoder) (Cho *et al.*, 2014)的提出，改善了單個 RNN (Recurrent Neural Networks) (Elman, 1990)長期記憶的不足，並提升了在自然語言處理領域各種任務的效果。但因輸出解碼過程在不同時間步所輸入的編碼向量相同，導致在轉換的過程中容易損失許多訊息，若提高向量大小則會導致計算量增加，而注意力機制的提出可以有效提高模型編碼的效率。

### 2.2 記憶網路 (Memory Network)

記憶網路(Memory Network, MemNN) (Weston, Chopra & Bordes, 2014)由Facebook 人工智慧實驗室所提出，目的在提高類神經網路模型長期記憶能力，應用於序列性資料上。如保存問答任務的先驗知識、聊天的語境訊息等。過往在處理序列性資料時，RNN 可以有有效的處理短期時間先後關係，每個時間步都會參考上一個時間步輸出的結果，但其只透過記憶單元儲存重要資訊，隨時間步推移更新記憶單元內容，對於長序列的訓練過程中可能會有梯度消失(gradient vanishing) 與梯度爆炸(gradient exploding) 的問題發生，造成 RNN 在長期記憶中表現不是很好。即使後來長短期記憶模型(Long Short-Term Memory, LSTM) (Hochreiter & Schmidhuber, 1997)相對提高了長期記憶能力，但對於更大的序列仍然有其限制存在。

記憶網路的限制在於訓練過程需要透過強監督的方式學習(Strong-Supervised Learning)，訓練用數據需要提供與查詢相關的標註句子，然而並非所有數據集都有支持

事實的標註，並不有利於將此模型應用到不同的數據集或不同任務上。端對端記憶網路(End-to-End Memory Networks, MemN2N)模型(Sukhbaatar, Szlam, Weston & Fergus, 2015)，在記憶網路模型的基礎上修改與完善，使其能以端對端的方式完成學習。透過弱監督方式(Weak-Supervise Learning) 即可完成訓練，有利模型的擴展並應用到不同的任務或資料集上。此模型利用軟性注意力機制(Soft Attention Mechanism) 來估計每條記憶與問題相關的程度，並使用相關性高的記憶計算出最後的輸出。

動態記憶網路(Dynamic Memory Networks, DMN)模型(Kumar *et al.*, 2016)，將大部份自然語言處理領域的任務視為問答任務的一種。該模型以記憶網路為基礎進行改善，可透過端對端學習完成訓練，應用於問答任務、情感分析以及詞性標註等。模型架構與記憶網路模型相似，主要由四個模組所組成，分別為輸入模組、問題模組、情景記憶模組(Episodic Memory Module) 與應答模組。與前述記憶網路的不同在於編碼方式。此模型採用門控循環單元模型(Gate Recurrent Unit, GRU) (Chung, Gulcehre, Cho & Bengio, 2014) 編碼，隨著時間步的推移更新隱藏狀態。相較於單純使用詞袋(Bags of word, BOW) 更可以表示出字詞之間的順序關聯。

在問答系統中加入知識庫(Knowledge Bases, KBs)可以有效的提高模型的知識儲存量，但其並不夠完整，無法支持不同類型的答案，由於數據的稀疏性，較難創建包含所有領域的 KB，不利於擴展到不同的領域。鍵值記憶網路(Key-Value Memory Networks) 模型(Miller *et al.*, 2016)使用鍵值(key-value) 的方式將文章中的編碼存取下來，架構基於端對端記憶網路模型，針對先驗知識的儲存提出不同方式編碼，應用於自然語言中問答的相關領域。

鍵值記憶網路模型與端對端記憶網路模型相似，最大的不同在於記憶的儲存方式。端對端記憶網路透過不同的嵌入矩陣對文本編碼，而鍵值記憶網路則透過鍵值的方式表示，以鍵記憶(key memory)與值記憶(value memory)形式儲存。鍵值記憶網路優點為在訓練網路之前，可先對先驗知識進行適合的編碼。即使是不同領域的知識，使用者也可選擇編碼方式，而不單純依賴於詞嵌入矩陣的訓練，在使用上有了更多的彈性。

遞歸實體網路(Recurrent Entity Networks, EntNet)模型(Henaff, Weston, Szlam, Bordes & LeCun, 2017)，記錄世界的實體與狀態於記憶中，當有新資訊輸入時，則根據輸入訊息更新相對應記憶單元，可應用於自然語言中的閱讀理解與問答任務中，其在 bAbI-10k 數據集與 Children's Book Test (CBT)數據集 single hop 訓練中，較更早提出之方法表現為優。

我們依照論文(Sukhbaatar *et al.*, 2015) (Henaff *et al.*, 2017)中之方法，以一千筆訓練資料進行實驗發現，最初記憶模型有效改善長期記憶關係，可在 bAbI 資料集中通過 16/20 項任務。但需要透過強監督方式進行訓練，並不有利於擴展應用。而弱監督訓練則只能通過 2/20 項任務，且錯誤率大幅增長。而端對端記憶網路模型透過弱監督方式訓練，相較於弱監督記憶網路，通過任務比例提昇，也大幅下降平均錯誤率。

而綜合前述論文所提供的數據，以一萬筆訓練資料為前提實驗，相較於前述以一千

筆資料訓練，模型相對通過更多的任務。端對端記憶網路模型通過了 17/20 項任務；動態記憶網路模型通過了 18/20 項任務。而首先通過所有任務的模型為遞歸實體網路模型，其平均錯誤率降低至 0.54%。如表 1 所示。

**表 1. 不同模型於 10k 數據量實驗結果**  
**[Table 1. Experimental results of different models with 10k data]**

Model	MemNN	MemN2N	DMN	EntNet
Mean Error	39.2	4.2	6.395	0.54
Failed Tasks(error>5%)	17	3	2	0

雖然在表 1 中 EntNet 在 10k 數據量，錯誤率小於 5% 的標準中通過所有任務，但其在部份任務中的錯誤率還是大於 4%。而且若是利用較少的 1k 的數據量進行訓練的話，正確度則會從原本的 99.5% 降到 89.1%。因此在較少數據的情況下，模型的準確性還有可以改進的空間。若是模型能提高少數據下訓練的效果，可以減少訓練時間，與資料收集的成本。而若是要應用到其他的資料量較少的情況，也能有比較好的效果。

### 2.3 多跳躍注意機制 (Multi-hop Attention)

多跳躍注意(Multi-hop Attention)機制於端對端記憶網路模型中所提出，透過不斷比對問題與記憶得出問題答案，這個過程模擬人類在推理過程的思考方式，後續模型透過不同的方式實現多跳躍機制，用以強化模型的推理能力。

問題簡化網路模型(Question Reduction Networks, QRN) (Seo, Min, Farhadi & Hajishirzi, 2017) 模型架構為 RNN 的一種，可有效處理短期與長期序列關係。透過多輪讀取機制，逐漸簡化問題，達到深入語意理解的效果，最後推理得出最終答案並轉化為自然語言輸出。此外 QRN 模型中所提出的公式允許在遞歸神經網路時間軸上並行化，提升訓練與推理部分的效率。

AOA Reader 模型(Attention-over-Attention) (Cui *et al.*, 2017)，應用於填空任務(Cloze-style questions)。與過往模型最大的不一樣在於透過不一樣的注意力機制組合，計算出權重預測最後的結果，而非使用單一一種注意力機制計算方法。模型透過雙向門控循環模型對先驗知識與問題進行編碼，將編碼向量點積相乘，經過 softmax 計算出詞彙的機率，此注意力機制的計算方法為許多模型通用方法，而此論文創新的地方為其不僅計算 document-to-query 的注意力數值，也計算 query-to-document 的注意力權重，最後利用兩者矩陣相乘得到最後注意力機制數值，並透過後續模型進行推理。

論文(Trischler *et al.*, 2016)提出了 EpiReader 神經網路模型，用以解決自然語言任務中的填空問題。EpiReader 模型分為兩個部分，第一部分為提取模組(Extractor)，通過淺層文本與問題的比對，提取出若干個問題的可能候選答案；第二部分為推理模組(Reasoner)，通過更深層的語意比較候選答案與問題之間的關聯。提取模組從大量可能性中篩選出小部分候選答案，而推理模組則處理更精確的推理匹配部分。

神經語意編碼器模型(Neural Semantic Encoders, NSE) (Munkhdalai & Yu, 2016)架構，在過往多輪讀取機制多為固定步數，但並非所有的問題需要相同推理的步數。有些問題只需要簡單的詞句比對即可得出結論，有些問題則需要複雜的語意理解與深度推理，因此 NSE 利用動態步數調整模型以解決此問題。

整理論文(Sukhbaatar *et al.*, 2015) (Henaff *et al.*, 2017) (Seo *et al.*, 2017)實驗數據以表格呈現，首先表 2 實驗在 bAbI 數據集上，以一千筆資料訓練，透過表格中可發現透過多跳躍機制可提高通過的任務數量或是降低平均錯誤率，不同多跳躍步數也會影響結果。

**表 2. QRN 與 MemN2N 模型不同 hop 實驗數據**  
[Table 2. Different hop experimental data of QRN and MemN2N models]

Model	MemN2N			QRN	
	1 hop	2 hop	3 hop	2r	3r
Mean Error	9.58	8.45	8.15	9.9	11.3
Failed Tasks(error>5%)	17	11	11	7	5

以兒童圖書測試數據(Children's Book Test, CBT)為實驗數據，表 3 整理幾種單跳躍與多跳躍模型實驗結果，目前效果最優為多跳躍模型，因此多跳躍機制成為目前研究領域的趨勢，而本論文研究也將嘗試以不同方式將多跳躍注意機制結合單跳躍模型。

**表 3. Single 與 Multi hop 不同模型於 CBT 數據及實驗結果**  
[Table 3. Different models of Single and Multi hop experimental results]

Model		Named Entities	Common Nouns
Single Pass	Kneser-Ney Language Model + cache	0.439	0.577
	LSTMs (context+query)	0.418	0.560
	Window LSTM	0.436	0.582
	EntNet (general)	0.484	0.540
	EntNet (simple)	0.616	0.588
Multi Pass	MemNN	0.493	0.554
	MemNN+self-sup	0.666	0.630
	EpiReader	0.697	0.674
	AoA Reader	0.720	0.694
	NSE	0.732	0.714

## 2.4 關係網路 (Relation Network)

關係網路(Relation Network) (Santoro *et al.*, 2017)目的在於透過加入對物件、實體或是語句之間的關係計算，提供更多訊息給後續推理模組進行推理。關係網路應用於圖像問答(Visual Question Answering, VQA)，使用簡單的模型來建構物體之間的聯繫，核心概念在於最終答案與成對的對象具有一定的關聯性，而問題也會影響對成對對象的查詢。其透過神經網路計算任意對象兩兩之間的潛在關係。

關係網路的優勢在於其架構簡單，使用彈性大，可將其插入於不同的模型裡，提高了模型推理的能力，可應用於關係推理相關任務上。前述論文在實驗中使用問答相關資料集做為驗證，在 bAbI dataset 二十個任務中通過了十八個，而在 Sort-of-CLEVR 中取得最優的結果，且超過人類所能達到的分數。

RelNet (Bansal, Neelakantan & McCallum, 2017)中將計算兩兩物件之間關係的概念帶入遞歸實體網路中，此論文將關係概念用來計算記憶與記憶之間的關聯。過往遞歸實體網路模型與記憶網路相關模型，記憶的儲存相互之間獨立，而 RelNet 模型透過關係計算將記憶彼此連結起來。計算記憶狀態儲存的公式與原模型相同，差別在多加上了計算不同記憶之間的關聯，並應用於最後的推理計算上。

遞歸關係網路(Recurrent Relational Networks, RRN)模型(Palm, Paquet & Winther, 2017)運用節點關係解決數獨的問題，在 9\*9 的數獨內共有 81 個節點，每個節點都需要考慮同一行、同一列與同個方框內的訊息，不能出現同樣的數字。此模型對每個節點初始化的狀態為  $\{h_1, h_2, \dots, h_{81}\}$ ，透過多層感知器(MLP) 計算每個節點之間的關聯，將計算出的關係數值相加，用以更新結點的狀態，每個節點的更新考慮上一個時間步的狀態、輸入以及關係數值。此模型除了應用於數獨上，也在 bAbI 數據集、Pretty-CLEVR1 表現優秀。

以(Cui *et al.*, 2017) (Trischler *et al.*, 2016)兩篇論文實驗所提供的資料為基礎，使用 bAbI 數據集中 10k 數據量訓練，並平均 20 項任務的錯誤率比較結果顯示於表 4。由實驗可以了解到，加入關係計算能有效的提升模型的訓練結果與計算。

**表 4. relation 相關模型比較**  
[Table 4. A comparison of relation models]

Method	Mean Error Rate (%)
RRN	0.46±0.77
RelNet	0.29
EntNet	9.7±2.6

## 2.5 預訓練模型 (Pre-trained Model)

近年來的語言模型研究使用大量文章預訓練(Pre-train)通用語言模型，然後再根據具體應用，用 supervised 的訓練資料，微調(Fine-tuning)模型，使之適用於具體應用，來提昇模

型的效能。

其中 BERT 模型(Devlin, Chang, Lee & Toutanova, 2018)以及後續發表，讓 BERT 更小，訓練更快的 Albert 模型(Lan *et al.* 2019)，被廣泛地應用於問答任務，且得到相當優異的成果。

這種結合預訓練模型再加上後續的微調訓練的方式，可以讓許多的自然語言處理任務得到極大幅度的效能提升，也讓我們可以用更小的資料就能訓練出極好的效果。

### 3. 研究方法 (Research Method)

記憶網路透過外部記憶的保存，強化長期記憶的能力，經過注意力機制找尋與問題相關的記憶槽，推理出對應的答案。記憶槽與記憶槽之間相互獨立運作，針對不同的實體保存相關訊息，對於需要多項記憶交互推理的複雜任務，推理模組較無法使用足夠的訊息輸出正確答案。本研究則透過記憶之間的關聯計算與多跳躍機制推理，嘗試提高模型記憶儲存與推理能力，並以問答相關任務作為驗證，模型需要具備語言理解以及推理的能力。接下來我們將介紹本論文整體模型的架構，並介紹各模組內的設計。

#### 3.1 模型架構 (Model Architecture)

本論文之模型架構以 EntNet 模型(Henaff *et al.*, 2017)為基礎，用固定大小記憶單元保存輸入數據的實體與相關屬性，記憶內容則隨著輸入句子即時更新。模型架構在記憶槽(memory slot)之間加上記憶關聯的計算與提取，保存於關聯槽(relation slot)內。原模型記憶槽與記憶槽之間相互獨立運作，但記憶與記憶之間應具有一定的關聯，透過關聯計算可將各記憶槽內容聯繫起來。模型主要可分為三個部分，分別為 Encoder 模組，負責將輸入的自然語言詞句編碼為向量的形式，以利電腦後續計算；動態記憶模組在每次句子輸入時，更新記憶槽內所儲存的資訊，再透過計算記憶槽彼此間的關聯來更新關聯槽，記憶槽大小與關聯槽大小相等；最後輸出模組根據問題，從記憶槽與關聯槽中推理出最後的答案。整體架構如下方圖 1 模型架構圖所示。相較 EntNet，在本架構中我們加入了 Relation memory 的部份，以嘗試透過結合記憶關聯計算，增加記憶間的聯繫。

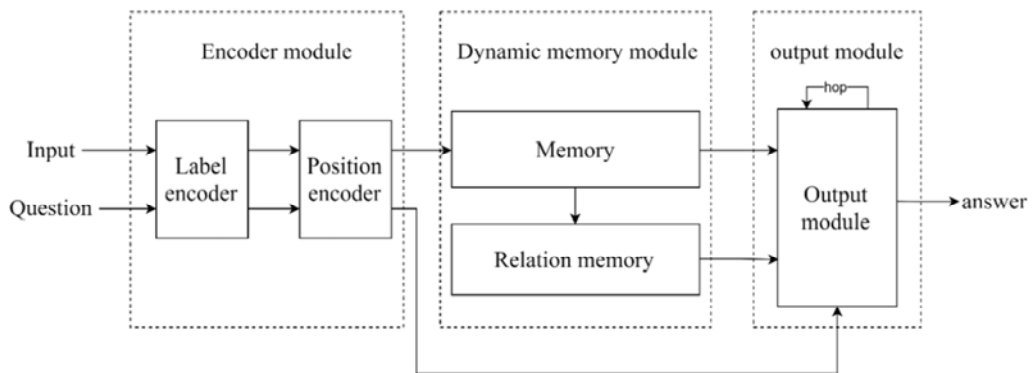


圖 1. 模型架構圖。

[Figure 1. Diagram of the model architecture]



### 3.2 Encoder模組 (Encoder Module)

此模型應用於問答任務上，所使用的數據皆為自然語言形式，自然語言無法直接輸入至電腦計算，因此在輸入至模型訓練前須先轉換為編碼的形式，以利後續的運算，此模組分為兩個步驟編碼，透過 Label encoding 初步將句子轉換為數字，再經過 Position encoding 給予字詞於整體句子的相對位置資訊。首先建立詞彙庫，將數據集中所有用到的詞彙對應到一個固定的數字，詞彙庫中詞彙量與數字量相等，不會新增多於欄位的詞彙。完成詞彙庫的建立後，將數據集根據詞彙庫轉換為數字的形式表示。範例如下所示：{}內為不同詞彙所對應的編號，[]為每個句子依照詞彙庫轉換為對應編號。

{hallway:1,John:2,the:3,to:4,went:5,:6}

John went to the hallway.→[2,5,4,3,1,6]

基本數值轉換後，雖然句子都以數字形式表示，但編碼並無相對應的意義，以上面例子為例，hallway 數值為 1、John 數值為 2，hallway 的兩倍為 John，這並無法解釋詞與詞之間的關係，所以這些數字將會再次轉換為模型訓練出來的向量，而數字是為了將相同的詞彙轉換為相同的向量。首先根據詞彙庫的大小，建立與詞彙量相等量的可訓練向量，每個詞彙有對應的向量，並在整體模型訓練的過程中一起更新向量數值，透過使用自然語言相關數據集，訓練詞彙對應的向量。

位置編碼 (Position encoding) 目的在於賦予字詞之間順序的關係，自然語言語意會根據詞彙的順序而有所不同，若是使用 BOW 的方式編碼，詞彙出現在句子任意位置對於編碼並沒有不同，但於實際語言相同詞彙於不同位置，對於語意理解可能會有很大程度的不一樣，如下方範例所示，相同用詞於不同位置所得出的語意相差甚大。

John likes Mary.≠Mary likes John.

本實驗位置編碼採用訓練的方式達成，透過 mask 的方法為語句加入順序關係，如下方公式(1)所示。 $\{e_1, \dots, e_k\}$ 為句子中每個詞彙的編碼向量， $\{f_1, \dots, f_k\}$ 是需要學習的 multiplicative mask，為可訓練的向量。使用這個 mask 的目的在於加入位置資訊。透過訓練的過程更新權重，當相同詞彙於不同的位置時，所乘上的 $f_k$ 向量也會有所不同。透過這樣的方式將位置的訊息加入至編碼中，最後將其加總表示整體句子的向量。

$$s_t = \sum_i f_i \odot e_i \quad (1)$$

### 3.3 動態記憶模組 (Dynamic Memory Module)

動態記憶模組由兩個部分組成，分別為記憶儲存槽與關係儲存槽，記憶槽以 key-value 的形式保存，key 保存實體、value 保存狀態，更新完記憶後再依據記憶與記憶之間的關聯更新關係槽內容，記憶槽與關係槽數量相等，架構如下方圖 2 所示。本架構與 EntNet 的差別為本研究加入了關係儲存槽 r。新加入部份在圖中以較粗之線條繪出。

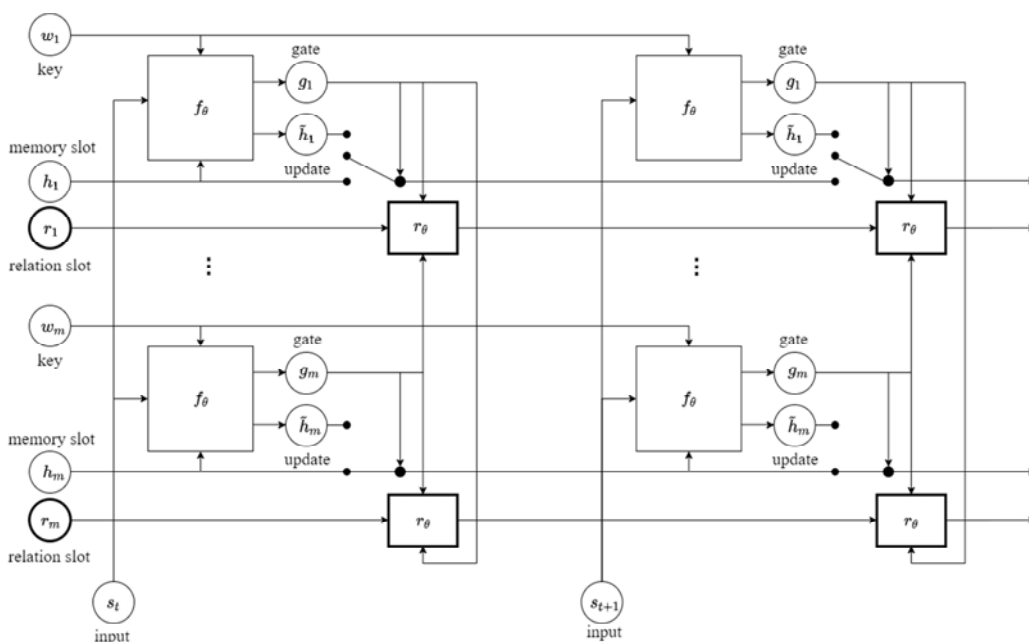


圖 2. 動態記憶模組架構圖

[Figure 2. Diagram of the dynamic memory module architecture]

數據編碼完後以向量形式表示每個句子 $s_t$ 。 $t$ 為不同時間步的句子，依照順序輸入至模型內更新記憶槽與關係槽。每個記憶槽由key和value組成，分別為 $w_i$ 和 $h_i$ ，以key-value的形式保存資訊。key負責保存實體、value負責保存狀態，範例如下方所示。範例中key保存了John這個實體，value保存了John所做的動作，每個記憶槽都有自己的key與value向量，透過輸入數據與key-value的比對可找到此次狀態更新應該更新於哪個記憶槽。

John went to hallway. => {key:John,value:went to hallway}

當每個句子輸入至模型內時，系統透過公式(2)計算句子與key、value之間的關係。 $\sigma$ 表示sigmoid activation function， $g_i$ 是gate，輸出數值將介於0~1之間，此數值為門控機制，用以決定更新與保存多少記憶內容。 $g_i$ 由 $w_j$ 和 $h_j$ 決定。前者表示與關鍵字的匹配程度，後者表示與memory內容的匹配程度。與此記憶槽實體越相關的語句，所計算出的數值會越高。公式(3)為RNN的計算公式，用以計算出輸入句子的內容。 $\tilde{h}_j$ 表示需要新增到已有的memory中的狀態值。 $\phi$ 可以是任意的activation function，實驗進行時使用的是PReLU。 $U$ 、 $V$ 、 $W$ 皆為可訓練權重，並且所有的gated RNN共享這些引數，於整體模型訓練時一起更新。公式(4)更新每個記憶槽 $h_j$ 內容，將原記憶槽內容門控與新記憶相加，透過門控數值控制更新的幅度。公式(5)用以遺忘不必要資訊，若是不斷將新記憶加入記憶槽內，向量數值將會越來越大，透過除上normalization數值，保持記憶向量數值範圍。

$$g_i \leftarrow \sigma(s_t^T h_j + s_t^T w_j) \quad (2)$$

$$\tilde{h}_j \leftarrow \phi(Uh_j + Vw_j + Ws_t) \quad (3)$$

$$h_j \leftarrow h_j + g_j \odot \tilde{h}_j \quad (4)$$

$$h_j \leftarrow \frac{h_j}{\|h_j\|} \quad (5)$$

模型中每個關係槽保存對應記憶與所有其他記憶的關係。更新完記憶槽將會得到此次輸入對於每個記憶的門控數值，將門控數值兩兩相乘計算彼此間的關聯，相同的門控計算加總於相同關係門控數值 $g_i^r$ ，如公式(6)所示。此公式目的在於將相同關聯對象保存在一起。其中 $g_i^m$ 、 $g_j^m$ 依據相同關聯對象，選擇相關的記憶槽。例如模型有 20 個記憶槽，記憶槽 1 與其他所有的記憶槽計算出 19 個關係，將這些關係保存於第一個關係槽。如此，後續推理時可直接從對應關係槽找出與其他記憶的關聯，如公式(7)所示。

$$g_{ij}^r = g_i^m g_j^m \sigma(\langle s_t, r_{ij} \rangle) \quad (6)$$

$$g_i^r = \sum_{i,j} g_{ij}^r \quad (7)$$

公式(8)計算關係更新內容，向量 A、B 為可訓練權重，根據原本關係槽內容 $r_{ij}$ 與輸入句子 $s_t$ 所需要更新的內容，最後以 PReLU 為 activation function。公式(9)為關係更新。將關係門控數值乘上新的關係內容，並加上原本的關係槽內容用以更新關係槽資訊。

$$\tilde{r}_{ij} \leftarrow PReLU(Ar_{ij} + Bs_t) \quad (8)$$

$$r_{ij} \leftarrow r_{ij} + g_{ij}^r \odot \tilde{r}_{ij} \quad (9)$$

### 3.4 輸出模組 (Output Module)

動態記憶模組更新完記憶槽 $h_i$ 與關係槽 $r_i$ ，將最後狀態保存給輸入模組推理使用。公式(10)將同個記憶槽 $h_i$ 與關係槽 $r_{ij}$ 向量並接在一起，並乘上可訓練權重 $C$ 計算出記憶 $m_i$ 。然後再透過注意力機制計算與 query 相關的 $p_i$ 數值，數值越高代表相關性越高，如公式(11)所示。

$$m_i = C[h_i; r_{ij}] \quad (10)$$

$$p_i = \text{Softmax}(q^T m_i) \quad (11)$$

將注意力數值乘上對應記憶，越相關記憶數值相對會越高如公式(12)所示，以保留與問題相關重要資訊。系統最後根據公式(13)推理出最後問題的答案，其中  $R$  跟  $H$  為參數矩陣。query 問題會依照訓練時的方式被編碼成  $k$  個維度的向量  $q$ 。本研究使用數據為問答任務，系統根據詞彙庫輸出最有可能的答案  $y$ 。

$$u = \sum_i p_i m_i \quad (12)$$

$$y = R\theta(q + Hu) \quad (13)$$

從第二章文獻探討可發現，多跳躍機制有助於提升模型推理能力，本研究嘗試將此概念加入推理模組，將上方記憶與注意力權重相乘加總的向量  $u$ ，與原 query 向量相加，作為新的 query 向量，重複公式(11)(12)計算，每多一次計算 hop 數增加 1，原本推理模組為 hop1，重複一次計算為 hop2，依此類推，如公式(14)所示。

$$q = q + u \quad (14)$$

### 3.5 討論 (Discussion)

本研究以 EntNet 模型為研究基礎，嘗試透過結合記憶關聯計算，增加記憶間的聯繫，而非不同記憶槽獨立運作。在 3.1 節中介紹整體模型架構。主要 Encode 模組、動態記憶模組以及輸出所組成。3.2 節介紹文字如何轉換為向量形式表示。從建立基本詞彙庫到訓練詞彙向量的過程。3.3 節中介紹動態記憶模組細節。負責記憶保存與更新的部分，除了原記憶模型的記憶槽外，將關係的計算加入模型內，使不同的記憶槽可透過關聯計算，計算彼此的關係，記憶間的關聯計算隨著記憶槽數量而快速增長，將其提取為同樣記憶數量的關聯槽，可降低權重與計算量。3.4 節為輸出模組的細節。當記憶模組將先驗知識保存後，輸出模組針對問題從記憶中取出相關的部分，並推理出最後的答案。研究方法中的關係計算與多跳躍的推理方法，可泛化應用到不同的記憶網路架構，或是具有記憶保存的架構的模型上。

## 4. 實驗 (Experiments)

本研究所有實驗選擇以 bAbI dataset 做為實驗驗證的數據集，此數據集為自 Facebook AI Research (FAIR)所提供的綜合閱讀理解和問答資料集。選擇此數據集驗證目的有四點，分別如下：

- (一) 數據集包含了二十種任務，可從不同面向測試模型的優勢與劣勢。
- (二) 為問答與自然語言理解常用數據集，有許多不同模型實驗數據可參考比較。
- (三) 包含英文、印地語與改組(人類不可閱讀) 等數據，可了解語言模型應用於不同自然語言之效果。
- (四) 於 20 項任務中提供 1k 資料量與 10k 資料量，可實驗數據量多寡對於模型學習的影響。本研究目標為於數據集 1k 的前提下，提升模型訓練效果。

以下實驗為求準確性，以交叉驗證方式，透過不同訓練數據做驗證，並平均多次訓練結果。實驗使用 1k 數據量訓練模型，並將 10k 數據切分多份 1k 檔案，透過多次實驗驗證改善效果。

#### 4.1 實驗一(多點跳躍訊息推理) (Experiment 1: Multi-hop Reasoning)

##### 實驗目的：

由前面的實驗及討論中，我們可以看出多跳躍針對複雜的問答推理，普遍相較於單跳躍對於推理結果效果更好，而 EntNet 模型屬於單跳躍模型，本實驗嘗試將多跳躍的概念應用於輸出模組中，嘗試增強遞歸神經網路推理能力。

##### 實驗內容：

本實驗將引入端對端記憶網路的多跳躍推理公式，選用此方法原因在於端對端網路與 EntNet 模型相似，都具備記憶單元保存資訊，其他模型架構方法差異較大。輸出模組負責推理答案，透過注意力機制找尋與此次問題相關的記憶，依據記憶內容推理出最終答案，而經過一次注意機制的計算為單跳躍，本實驗嘗試增加跳躍數量。如 3.4 小節中的公式(14)，注意力機制計算出的數值與計算所使用的 query 相加，做為新的 query 再次與記憶做注意力機制的計算，每多做一次跳躍數增加 1，實驗將比較雙跳躍與原先單跳躍的差異。實驗結果整理於表 6 內之 Multi hop 欄位。

由實驗數據中可以看出，增加跳躍數量並沒有增加模型的準確率，甚至部分的準確率相較於原模型有下降的趨勢，平均錯誤率反而上升。分析訓練出的模型於訓練資料、測試資料的表現，可以看出有過擬合的趨勢，複雜化推理模組無法提升效果。推測為記憶模組所保存的資訊不足，無法提供足夠的資訊給與推理模組進行後續的推理。因此設計實驗二透過記憶關聯的計算，與關係槽的保存提升模型外部記憶保存的能力，嘗試保存更關的資訊是否能提升模型的效果。

#### 4.2 實驗二(記憶關聯) (Experiment 2: Memory Relation)

##### 實驗目的：

此實驗主要應用於 EntNet 模型的動態記憶模組，根據實驗一的實驗結果，可發現複雜化推理模組無法提升推理能力，因此實驗二將關聯計算加入記憶之間。相較於原本記憶分別獨立保存訊息，記憶關聯的計算能將相關記憶串連起來。如同人類記憶保存並非把所有部分完全獨立，透過聯想可聯繫到不同的想法或記憶。本實驗額外保存記憶與記憶的間的關聯，用以增加推理模組可用訊息，增加模型推理能力。

##### 實驗內容：

本實驗將記憶槽兩兩根據公式(6)計算出關係門控數值，用以決定此次數據輸入對於關係槽更新多寡。實驗記憶槽數量與原模型相同，使用 20 個記憶槽保存重要資訊。另外額外加入關係槽用以保存對應記憶槽的關係，如第一個記憶槽與其他所有記憶槽關聯計算，保存於第一個關係槽，關係槽數量與記憶槽數量相等，確保關係保存不會隨記憶槽增長而大量增加。關聯計算透過  $C_m$  2 排列組合計算，如公式(15)所示。20 個記憶槽計算出

190 個關係，再將 190 個關係分別保存到對應的關係槽內。實驗結果如表 6 中之 Relation slot 欄位所示。

$$\text{number of relation} = C_2^m \quad (15)$$

bAbI 數據集於不同任務有不同的推理難度，有些任務需要結合多項先驗知識交叉推理才能得出答案。從實驗數據中可以看出，透過關聯計算能有效降低平均錯誤。相較於原先記憶獨立保存，此方法可以更有效地從數據中學習詞句間的關聯。但也並非所有任務都有明顯改善。任務 2 改善效果最為明顯，數據特性是找到兩項支持事實的句子，才能推理出問題的答案，而關聯的計算剛好是兩兩記憶槽的計算，效果顯著於提升任務 2。但任務 3 更多的支持事實句子卻無法提升。推論為數據集太小以及關聯計算的方法的影響。

另於表 5 比較保存所有關聯計算(如 RelNet 模型即是採用此方法)，與關聯提取兩種方法。對於權重的使用量，權重數量越多代表所需要 GPU 所需要的計算量越大。本實驗中 20 個記憶槽將會計算出 190 個關係，此實驗將 190 個關係數值提取於 20 個關係槽保存，目的只保存重要的資訊。如此可以大為減少模型權重的數量 6 萬個，較原使用所有關聯的權重數量降低了 26.8%。實驗結果的數據可以發現在不同任務提升效果不同，也有部分準確率是些微下降，但整體仍以提升為主。

**表 5. 保存所有記憶關聯與提取方法權重數量比較**  
**[Table 5. A comparison between all relation method and relation slot method]**

Task	All relation method	Relation slot
Task 1: Single Supporting Fact	110000	80000
Task 2: Two Supporting Facts	112900	82900
Task 3: Three Supporting Facts	113400	83400
Task 4: Two Argument Relations	109400	79400
Task 5: Three Argument Relations	115200	85200
Task 6: Yes/No Questions	113400	83400
Task 7: Counting	115100	85100
Task 8: Lists/Sets	115100	85100
Task 9: Simple Negation	111100	81100
Task 10: Indefinite Knowledge	111500	81500
Task 11: Basic Coreference	111600	81600
Task 12: Conjunction	110400	80400
Task 13: Compound Coreference	111600	81600
Task 14: Time Reasoning	111700	81700

Task 15: Basic Deduction	109700	79700
Task 16: Basic Induction	109500	79500
Task 17: Positional Reasoning	111100	81100
Task 18: Size Reasoning	110700	80700
Task 19: Path Finding	113200	83200
Task 20: Agent's Motivations	113600	83600
Sum of all task parameters	2240200	1640200
Mean parameters	112010	82010

### 4.3 實驗三(自我記憶關聯) (Self memory Relation)

#### 實驗目的：

實驗二中將計算出的 190 個關係提取為與記憶槽數量相等的關係槽，透過這樣的方式輸出模組不需要將所有關係都計算過，在動態記憶模組進行保存時，即可篩選出重要的關聯資訊進行保存，並非所有關係數值都需要被保存，輸出可以只專注於重要的資訊推理答案。此實驗嘗試將記憶關聯直接更新於原記憶槽內，而不另外透過關係槽保存關係資訊，實驗是否透過記憶的自我關聯更新，即可提升記憶槽保存內容的品質。

#### 實驗內容：

關聯計算方法同實驗二，差別在於實驗三更新的目標，為記憶槽本身所保存的內容，原輸入關聯槽的部分改成記憶槽本身，輸出所更新的目標也是記憶槽。如公式(16)~(18)所示，計算此次輸入句子在兩兩記憶槽間的關係，方法與實驗二相似，而公式(19)透過門控數值決定此次更新的多寡，而更新的目標是記憶本身。實驗結果如表 6 中之 Self memory 欄位所示。

$$g_{ij}^r = g_i^m g_j^m \sigma(< s_t, h_i >) \quad (16)$$

$$g_i^r = \sum_{i,j} g_{i,j}^r \quad (17)$$

$$\tilde{r}_i \leftarrow PReLU(Ah_i + Bs_t) \quad (18)$$

$$h_i \leftarrow h_i + g_i^r \odot \tilde{r}_i \quad (19)$$

表6. 原模型與多跳躍、關係計算、自我關聯更新實驗結果(錯誤率)  
 [Table 6. Error rate of different models]

Task	Original model	Multi hop (hop2)	Relation slot	Self memory
Task 1: Single Supporting Fact	0.00%	0.00%	0.00%	0.00%
Task 2: Two Supporting Facts	20.80%	28.40%	11.60%	52.30%
Task 3: Three Supporting Facts	58.70%	56.70%	62.90%	62.10%
Task 4: Two Argument Relations	0.10%	0.20%	0.00%	0.00%
Task 5: Three Argument Relations	1.20%	1.20%	1.40%	17.20%
Task 6: Yes/No Questions	3.60%	3.50%	1.90%	11.40%
Task 7: Counting	10.10%	10.10%	6.90%	23.40%
Task 8: Lists/Sets	1.30%	2.20%	1.70%	9.10%
Task 9: Simple Negation	0.40%	0.00%	0.00%	35.60%
Task 10: Indefinite Knowledge	0.50%	3.70%	0.80%	3.80%
Task 11: Basic Coreference	8.90%	8.00%	4.20%	7.50%
Task 12: Conjunction	0.00%	0.00%	0.00%	0.60%
Task 13: Compound Coreference	5.60%	5.60%	6.20%	5.80%
Task 14: Time Reasoning	20.50%	21.30%	20.60%	55.90%
Task 15: Basic Deduction	5.10%	29.70%	0.00%	45.80%
Task 16: Basic Induction	50.00%	50.70%	51.00%	51.20%
Task 17: Positional Reasoning	41.20%	39.00%	37.70%	39.40%
Task 18: Size Reasoning	8.00%	7.60%	6.20%	8.50%
Task 19: Path Finding	87.80%	86.80%	85.30%	87.60%
Task 20: Agent's Motivations	0.90%	0.20%	0.90%	2.90%
Mean Error	<b>16.24%</b>	<b>17.74%</b>	<b>14.96%</b>	<b>26.00%</b>
Failed Tasks(error>5%)	<b>11</b>	<b>11</b>	<b>9</b>	<b>15</b>

相較於實驗二加入關聯計算的提升，此實驗平均準確率反而下降不少。推論為關聯計算雖能提升模型推理效果，但若是直接更新記憶槽本身，反而會造成記憶保存的效果下降，目前仍是將兩者分開保存效果較好。但根據表 7 權重的數量比較，可以發現自我記憶關聯更新可以下降不少權重運算，單個任務可下降 1 萬權重量，所有任務為 20 萬權重。



**表7. 實驗二關聯槽方法與實驗三自我關聯更新權重比較**  
**[Table 7. A comparison between relation slot and self memory update]**

Task	Relation slot	Self memory update
Task 1: Single Supporting Fact	80000	70000
Task 2: Two Supporting Facts	82900	72900
Task 3: Three Supporting Facts	83400	73400
Task 4: Two Argument Relations	79400	69400
Task 5: Three Argument Relations	85200	75200
Task 6: Yes/No Questions	83400	73400
Task 7: Counting	85100	75100
Task 8: Lists/Sets	85100	75100
Task 9: Simple Negation	81100	71100
Task 10: Indefinite Knowledge	81500	71500
Task 11: Basic Coreference	81600	71600
Task 12: Conjunction	80400	70400
Task 13: Compound Coreference	81600	71600
Task 14: Time Reasoning	81700	71700
Task 15: Basic Deduction	79700	69700
Task 16: Basic Induction	79500	69500
Task 17: Positional Reasoning	81100	71100
Task 18: Size Reasoning	80700	70700
Task 19: Path Finding	83200	73200
Task 20: Agent's Motivations	83600	73600
Sum of all task parameter	1640200	1440200
Mean parameter	82010	72010

#### 4.4 實驗總結 (Experiment summary)

前面的三個實驗中主要探討兩個方向：準確率與權重計算量。從準確率方面來看，實驗一於輸出模組中做更動，透過重複性的注意力機制計算，嘗試提升複雜任務的推理能力。而實驗一的實驗結果平均錯誤率反而略為提升，推論為記憶保存的內容不足以支撐複雜的推理過程。

實驗二於動態記憶模組中做改善。透過記憶間的關聯，計算連結步動記憶間的關係。

從實驗數據中可看出實驗二的改善效果較為明顯，特別是任務 2 的準確率大幅提升。

實驗三效果下降最多。將關聯計算與本身保存的記憶同時更新於同個記憶槽，反而造成模型整體效果下降。推論將關聯計算更新記憶槽，會造成記憶保存的混亂。目前方法仍是需要分開保存關聯資訊與記憶本身，但也不代表記憶的關係自我更新不可行，而是需要詳細研究記憶槽與關聯槽的內容與特性，從而找出更好的記憶保存方法。

從權重計算量方面來看，實驗一權重使用量最少。因模型尚未加進記憶關聯的計算，且因模型重複使用相同注意力機制重複計算，權重用量與原模型差異不大。

而實驗二關聯提取與所有關聯計算比較，權重下降幅度最多，所有任務整體下降了 60 萬權重數量，較實驗一下降了 26.8% 的權重量。實驗三雖準確率不高，但相較實驗二整體任務又下降了 20 萬權重。

## 5. 結論 (Conclusions)

本論文透過關係的計算使記憶內不同記憶槽具有關聯，如同人類記憶中不同實體並非單獨保存各自的訊息。例如實體的訊息或屬性可以聯想到其他實體或事件。關係的概念首先由 Google Deepmind 團隊於論文(Santoro *et al.*, 2017)中所提出，應用於圖像問答任務 (Visual Question Answering, VQA)，計算兩兩物體間的關係。而記憶網路的概念目的在於透過不同的記憶保存與推理方式，提升模型的長期記憶能力，RelNet 模型首先將關係的概念帶入記憶網路中，提升模型的準確性，但其缺點也很明顯：大量的提高模型的權重與計算量。本論文提出關係提取的概念可以大量減少權重的計算量。

實驗中所採用的問答任務使用 20 個記憶槽，關係的縮減從 190 個關係提取到 20 個關係槽中，即使是小型任務也可以發現計算量大為下降。而在越大型自然語言任務所運用的記憶槽數量相對越高，其中兩兩相對的關係計算也會大量增長。將關係提取出重要資訊於關係槽內，可以大量減少記憶儲存所需要占用的大小，以及輸出模組所需要推理的計算量。

透過實驗二實驗結果可以看出整體準確率可以有效的提升，而這樣的方法不侷限於實驗所使用的 EntNet 模型，也可以運用於不同記憶網路架構內。而非記憶網路模型也可以用同樣的概念計算物體、數據、詞句以及時間上的關係。

以本論文為基礎，未來還可朝關係計算方法進行改善。從實驗二中可以看出不同任務提升的效果不同，其中任務 2 提升效果最為明顯。與任務本身的特性有關，透過更多關聯計算或許可提升其他任務的準確率。本研究的關係計算所採用的是兩兩記憶槽的計算，但現實世界不同的實體關係可以是兩個、三個或是群體間具有一定的關聯，例如籃球、羽毛球與足球三者皆屬於球類。bAbI 數據集中的任務 3 也需要需要更多先驗知識交互推理。未來若是關聯計算能帶入群組關聯計算，增強記憶間的連結性，應能再次提升模型記憶保存與推理能力。

本論文所改善的部分皆落於記憶保存與推理的部分，編碼的部分應能透過預訓練的方式改善。近幾年的語言模型研究多為預訓練(Pre-train)與微調(Fine-tuning)的方法，透過

大量文本資料來訓練自然語言詞句的關係。透過未標註的大量資料訓練，使編碼的向量可以更準確的表示詞句的意思。

本研究的實驗編碼方式都是與整體模型一起訓練，包含編碼、動態記憶模組以及推理模組的權重，數據量的不足較無法深入學習詞句意涵，而目前網路文本資料量大，未來可將模型的 Encoder 模組經過預訓練，提升編碼效果，或是針對編碼方式做改進，提高整體模型預測的效果。

### 參考文獻(References)

- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.
- Bansal, T., Neelakantan, A., & McCallum, A. (2017). RelNet: End-to-End Modeling of Entities and Relations. In arXiv preprint arXiv:1706.07179.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., ... Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724-1734. doi: 10.3115/v1/D14-1179
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. In arXiv preprint arXiv:1412.3555.
- Cui, Y., Chen, Z., Wei, S., Wang, S., Liu, T., & Hu, G. (2017). Attention-over-Attention Neural Networks for Reading Comprehension. In *Proceedings of the 55th Annu. Meet. Assoc. Comput. Linguist, 1*, 593-602. doi: 10.18653/v1/P17-1055
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In arXiv preprint arXiv:1810.04805.
- Elman, J. L. (1990) Finding structure in time. *Cogn. Sci.*, 14(2), 179-211. doi: 10.1016/0364-0213(90)90002-E
- Henaff, M., Weston, J., Szlam, A., Bordes, A., & LeCun, Y. (2017). Tracking the World State with Recurrent Entity Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, 9(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735
- Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., ... Socher, R. (2016). Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 48, 1378-1387.

- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In arXiv preprint arXiv:1909.11942.
- Miller, A., Fisch, A., Dodge, J., Karimi, A.-H., Bordes, A., & Weston, J. (2016), Key-Value Memory Networks for Directly Reading Documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1400-1409. doi: 10.18653/v1/D16-1147
- Munkhdalai, T., & Yu, H. (2016). Reasoning with Memory Augmented Neural Networks for Language Comprehension. In arXiv preprint arXiv:161006454.
- Palm, R. B., Paquet, U., & Winther, O. (2017). Recurrent Relational Networks. In arXiv preprint arXiv: 171108028 Cs.
- Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P., ... Lillicrap, T. (2017) A simple neural network module for relational reasoning. In arXiv preprint arXiv:170601427.
- Seo, M. J., Min, S., Farhadi, A., & Hajishirzi, H. (2017). Query-Reduction Networks for Question Answering. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.
- Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-End Memory Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, 2, 2440-2448.
- Trischler, A., Ye, Z., Yuan, X., Bachman, P., Sordani, A., & Suleman, K. (2016). Natural Language Comprehension with the EpiReader. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 128-137. doi: 10.18653/v1/D16-1013
- Weston, J., Bordes, A., Chopra, S., & Mikolov, T. (2016). Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In *Proceedings of the ICLR2016*.
- Weston, J., Chopra, S., & Bordes, A. (2014). Memory Networks. In arXiv preprint arXiv:14103916.