國立臺灣科技大學
資訊工程系

碩士學位論文

深度神經網路於中文斷詞之研究

Research On Chinese Word Segmentation with Deep Neural Network

研究生：吳澤鑫
學號：M10615090
指導教授：陳冠宇 博士

中華民國 112 年 01 月 09 日

M10615090

碩士學位論文指導教授推薦書

Master's Thesis Recommendation Form

系所： Department/Graduate Institute	資訊工程系 Department of Computer Science and Information Engineering
姓名： Name	吳澤鑫 WU, TSE-HSIN
論文題目： (Thesis Title)	深度神經網路於中文斷詞之研究 Research On Chinese Word Segmentation with Deep Neural Network

係由本人指導撰述，同意提付審查。

This is to certify that the thesis submitted by the student named above, has been written under my supervision. I hereby approve this thesis to be applied for examination.

指導教授簽章：
Advisor's Signature

共同指導教授簽章（如有）：
Co-advisor's Signature (if any)

日期：
Date(yyyy/mm/dd) 2023 / 1 / 9

M10615090

碩士學位考試委員審定書

Qualification Form by Master's Degree Examination Committee

系所： Department/Graduate Institute	資訊工程系 Department of Computer Science and Information Engineering
姓名： Name	吳澤鑫 WU, TSE-HSIN
論文題目： (Thesis Title)	深度神經網路於中文斷詞之研究 Research On Chinese Word Segmentation with Deep Neural Network

經本委員會審定通過，特此證明。

This is to certify that the thesis submitted by the student named above, is qualified and approved by the Examination Committee.

學位考試委員會

Degree Examination Committee

委員簽章：

Member's Signatures

指導教授簽章：

Advisor's Signature

共同指導教授簽章（如有）：

Co-advisor's Signature (if any)

系所（學程）主任（所長）簽章：

Department/Study Program/Graduate Institute Chair's Signature

日期：

Date(yyyy/mm/dd) 2023 / 1 / 9

摘要

隨著 Transformer 架構的提出，如 BERT、GPT-2…等等，這些預訓練於大量文本上的模型透過微調下游任務的方式在自然語言處理領域中蓬勃發展。在中文斷詞領域中，也將資料集的評估分數推升至 F1 分數 97 分之超高分標準，但對於未知詞的問題上卻無法得到很好的解決。在此本論文針對中文的斷詞系統，提出一個結合監督式與非監督式方法的斷詞模型，提供一個可供參考之訓練框架，並且藉由此方式達到非監督式模型輔助監督式模型的效果，期望讓模型對於未知詞的處理能擁有更好的適應能力；在實驗中比較傳統 Word2Vec 預訓練字向量模型和目前主流的 Transformer 預訓練模型，並且探討目前中文斷詞領域，Word2Vec 模型之短處以及非監督式模型之架構改進方式，最後與目前常用套件進行效能上的比較。

Abstract

With the introduction of the Transformer architecture, such as BERT, GPT-2, etc., these pre-trained models on a large amount of text have flourished in the field of natural language processing by fine-tuning downstream tasks. In the field of Chinese word segmentation, the evaluation score of the data set has also been pushed up to the ultra-high score standard of F1 score of 97 points, but the problem of unknown words cannot be solved well. In this paper, a word segmentation model combining supervised and unsupervised methods is proposed for the Chinese word segmentation system. Provide a training framework for reference, and in this way achieve the effect of the unsupervised model assisting the supervised model, expecting the model to have better adaptability to the processing of unknown words; compare the traditional Word2Vec pre-training in the experiment The word vector model and the current mainstream Transformer pre-training model, and discuss the current field of Chinese word segmentation, the shortcomings of the Word2Vec model and the improvement of the structure of the unsupervised model, and finally compare the performance with the current commonly used packages.

致謝

首先，我要感謝的是我的家人們的支持與體諒，讓我在研究所五年半的時間裡得以繼續堅持下去，也感謝女友在精神上對我的加油打氣，陪伴我度過那些徬徨迷茫的日子。我也要特別感謝指導教授陳冠宇老師，在我有問題的時候總是不厭其煩的替我解答，在我遇到挫折時總是給我鼓勵加油，並時時督促我的實驗進度不要落後。另外，我也要感謝口試委員蘇明祥教授以及曾厚強教授，提點我許多論文中的不足之處，並提供了很多想法供我參考。

最後，我也要感謝實驗室的學長們：饒宗翰、湯京祐、施孟寰、高凱昱、易洋；同學們：羅上堡、吳政育、顏笠峰；學弟們：徐子杰、簡靖岳、楊志穎、張天佑…等，因為有了你們，讓我在研究所求學的時間裡給予許多心靈上的鼓勵與安慰，並且在畢業後也願意與我探討實驗上的問題，陪伴我度過實驗沒有成果的低潮時期。

最後感謝所有幫助及鼓勵過我而因篇幅未能提及的朋友們，因為你們心中有我，我才能走到最後這個階段。最後，願心中此份喜悅能與你們分享。

目錄

第 1 章　緒論 1
1.1　研究目的及動機 1
1.2　論文大綱 2

第 2 章　相關研究 3
2.1　基礎神經網路 3
2.1.1　卷積神經網路 3
2.1.2　長短期記憶體 4
2.1.3　Transformer 6
2.2　預訓練詞向量 11
2.2.1　詞向量表示法 11
2.2.2　連續詞袋模型和跳字模型 12
2.2.3　全局向量表示法 14
2.2.4　ELMo 15
2.2.5　OpenAI GPT 16
2.2.6　BERT 18
2.3　監督式學習斷詞標註方式 20
2.3.1　四分類法 20
2.3.2　六分類法 21
2.4　本論文相關斷詞模型 21
2.4.1　長詞優先斷詞 21
2.4.2　監督式學習方法 22
2.4.2.1　隱藏式馬可夫模型 22
2.4.2.2　最大熵馬可夫模型 24
2.4.2.3　條件式隨機域 27
2.4.2.4　雙向長短期記憶模型 28
2.4.2.5　堆疊卷積神經網路結合條件式隨機域 30
2.4.3　非監督式學習方法 33
2.4.3.1　基於分段語言模型的非監督斷詞法 33
2.4.3.2　使用雙向語言模型之非監督斷詞法 36

第 3 章　結合非監督式與監督式學習之斷詞方法 41
3.1　動機以及目的 41
3.2　結合非監督式與監督式學習之斷詞框架 42

第 4 章　實驗設定 43
4.1　資料集 43
4.2　評估方式 43
4.3　實驗設定 45
4.3.1　預訓練字向量 45
4.3.2　基礎實驗模型設定 45

第 5 章　實驗結果與討論 47
5.1　基礎系統 47
5.2　結合非監督與監督式學習之斷詞模型實驗 48
5.3　CNN 與 BiLSTM 模型之探討 49
5.4　雙向語言斷詞模型之探討 51
5.5　常用套件比較 55
5.6　基礎模型優缺點 56

第 6 章　結論與未來展望 57

參考文獻 58

圖目錄

FIGURE 1 CNN 示例圖 3
FIGURE 2 LSTM 架構圖 5
FIGURE 3 BI–LSTM 架構圖 6
FIGURE 4 TRANSFORMER 架構 7
FIGURE 5 SCALE DOT-PRODUCT ATTENTION 8
FIGURE 6 MULTI-HEAD ATTENTION 10
FIGURE 7 CBOW 模型與 SKIP-GRAM 模型 12
FIGURE 8 ELMO 模型架構圖 15
FIGURE 9 OPENAI-GPT 模型架構圖 17
FIGURE 10 BERT 模型架構圖 18
FIGURE 11 BERT 輸入向量示意圖 20
FIGURE 12 HMM 節點依賴圖 23
FIGURE 13 MEMM 節點依賴圖 25
FIGURE 14 標註偏移問題範例 26
FIGURE 15 CRF 節點依賴圖 27
FIGURE 16 BI–LSTM 斷詞架構圖 29
FIGURE 17 STACKED–CNN-CRF 模型架構圖 31
FIGURE 18 SLM 具體斷詞流程 35
FIGURE 19 模型架構圖 42

表目錄

表格 1　歧義性範例 1
表格 2　字元「中」可以出現在詞的任意位置 21
表格 3　正向長詞優先法與反向長詞優先法比較 22
表格 4　字結合詞特徵關係圖 32
表格 5　資料集統計 43
表格 6　混淆矩陣 44
表格 7　字向量訓練參數設定表 45
表格 8　基礎模型於 4 個資料集之 F1 分數 47
表格 9　U+BERT 實驗結果 48
表格 10　U+BERT OOV RECALL RATE 49
表格 11　標註錯誤範例 49
表格 12　基準模型數據 50
表格 13　去除 AS 資料集中文字元後之數據 50
表格 14　去除 AS 資料集英文、數字及標點符號之數據 50
表格 15　後處理後之數據 51
表格 16　遞迴結構測試於 AS 資料集 52
表格 17　遞迴結構測試於 CITYU 資料集 53
表格 18　遞迴結構測試於 MSR 資料集 53
表格 19　遞迴結構測試於 PKU 資料集 53
表格 20　資料集字詞統計表 54
表格 21　BERT 架構之 UCWS 54
表格 22　常用套件比較表 56

第1章 緒論

1.1 研究目的及動機

在中文自然語言處理領域中，如：機器翻譯[1, 2]、問答系統[3, 4]、自動摘要[5, 6]、語音辨識[7, 8]…等等，有些任務必須經過中文斷詞的處理才能進行下一步的處理，如：機器翻譯、資訊檢索[9]、詞性標記[10]。中文不像英文在每個詞之間會有空格區隔出每個詞。因此，自動中文斷詞[11, 12]在中文自然語言處理之前處理上是非常重要的一個步驟。若是中文斷詞的效果不好，可能會間接影響到下游任務的結果。

所謂的「中文斷詞」就是將一連串的中文「字串」轉換成「詞串」的組合。例如：「今天天氣很好」這個句子透過中文斷詞後變成「今天／天氣／很／好」。傳統上，中文斷詞主要可能遇到以下兩種問題：一是「歧義性問題」（Ambiguity），二是「未知詞」（Unknown Word)。歧義性問題就是同一字串在不同文章中會有不同的斷詞結果，例如表格 1 的例子。

表格 1　　歧義性範例

只有你才能勝任	只有／你／才／能／勝任
你才能非凡	你／才能／非凡

而未知詞則是指字典或訓練文章中未出現過的詞，可能是某個詞的簡稱、諧音…等，這些詞語在 PTT 文章中尤其常見，例如：「呱張」、「芭比 Q 了」。由於現今網路時代的盛行，時常衍生出許多新興的詞語，在利用辭典比對的斷詞系統上，很難將這類新興詞語完整的切分出來，因此若想要提升中文斷詞的正確性，斷詞系統就得必須能夠處理未知詞的問題。

1.2 論文大綱

本論文共分為六章：第一章為緒論，概述本論文研究動機及方向；第二章為相關研究整理，包含基礎神經網路、預訓練詞向量以及監督式和非監督式方法；第三章為本論文所提出之方法；第四章為實驗的設定和介紹資料集及其評估方式；第五章為實驗結果比較及探討；第六章為結論。

第2章 相關研究

2.1 基礎神經網路

2.1.1 卷積神經網路

卷積神經網路（Convolution Neural Networks，CNN）是一種前饋神經網路，在 1986 年反向傳播算法（Backpropagation）中提出[13]，由於當時的硬體能力尚未足夠成熟，直至 2006 年由 Hinton [14]發揚光大。CNN 的構造相似於模仿人類大腦的認知方式，例如：辨識一個圖像，會先注意到顏色鮮明的點、線、面，之後將它們構成一個個不同的形狀(眼睛、鼻子、嘴巴...等，這種抽象化的過程就是 CNN 演算法建立模型的方式。CNN 就是由點的比對轉成局部的比對，透過一塊塊的特徵研判，逐步堆疊綜合比對結果，就可以得到比較好的辨識結果。CNN 在電腦視覺（Computer Vision）[15, 16]領域上非常受到歡迎，因為它在處理圖像任務上能利用較少的參數去達到非常好的成績。

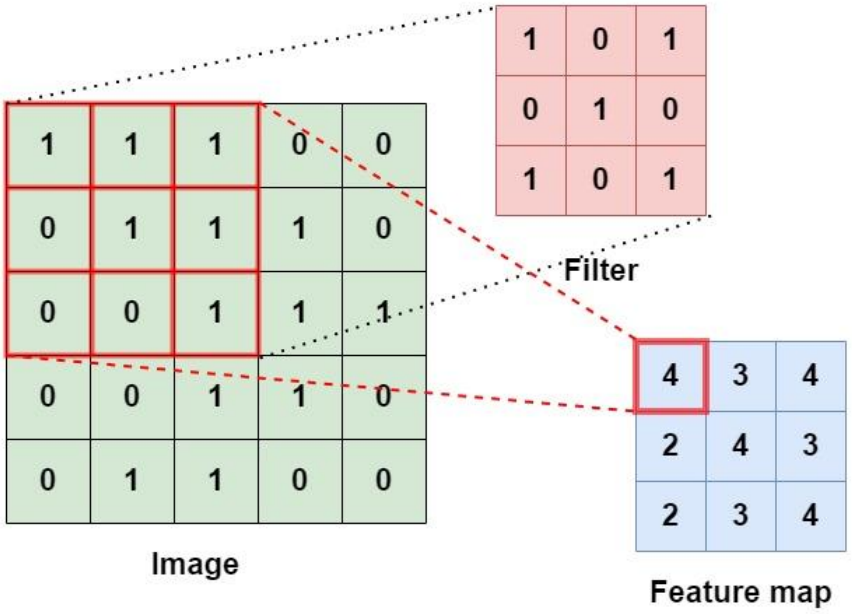

Figure 1 CNN 示例圖

以圖像的每一點為中心，取周遭$N \times N$格的點構成一個面（N稱為 Kernel Size，$N \times N$的矩陣權重稱為「卷積核」），每一格給予不同的權重，計算加權總和，當作這一點的輸出，再移動至下一點以相同方式處理，至圖像的最後一點為止，這就是 CNN 的卷積層（Convolution Layer），如 Figure 1。藉由給予卷積核不同的權重組合，就可以偵測形狀的邊、角，也有去除噪音（Noise）及銳化（Sharpen）的效果，萃取這些特徵當作辨識的依據。

2.1.2 長短期記憶體

遞迴神經網路（Recurrent Neural Networks，RNN）在解決一些語音轉文字或翻譯的領域上，不論在研究還是實務上的應用都已經展現相當不錯的效果。然而，這些好的效果中沒有一個是只透過簡單的 RNN 所達成的，絕大部分都會再改良模型當中的神經單元，變成更有效率的 RNN 改良版本，其中最有名的就是德國科學家 Hochreiter 和 Schmidhuber 於 1997 年提出的長短期記憶模型 (Long Short-Term Memory， LSTM)[17]以及 2014 年由韓國科學家 Kyunghyun Cho…等人所提出的門控循環單元（Gated Recurrent Unit，GRU）[18]。

長短期記憶網路 LSTM 是一種時間遞迴的神經網路，改良了傳統的 RNN 很難捕捉到長期的記憶的缺點，以及數學上所產生的梯度消失問題使得長時間的記憶會被短時間的記憶所隱藏。LSTM 創新的地方在於它在神經單元中加入了遺忘閥（Forget Gate）、更新閥（Update Gate）以及輸出閥（Output Gate）三個步驟，進而大幅提高了傳統 RNN 在長期記憶的表現並且解決了傳統 RNN 梯度消失的問題[19]。

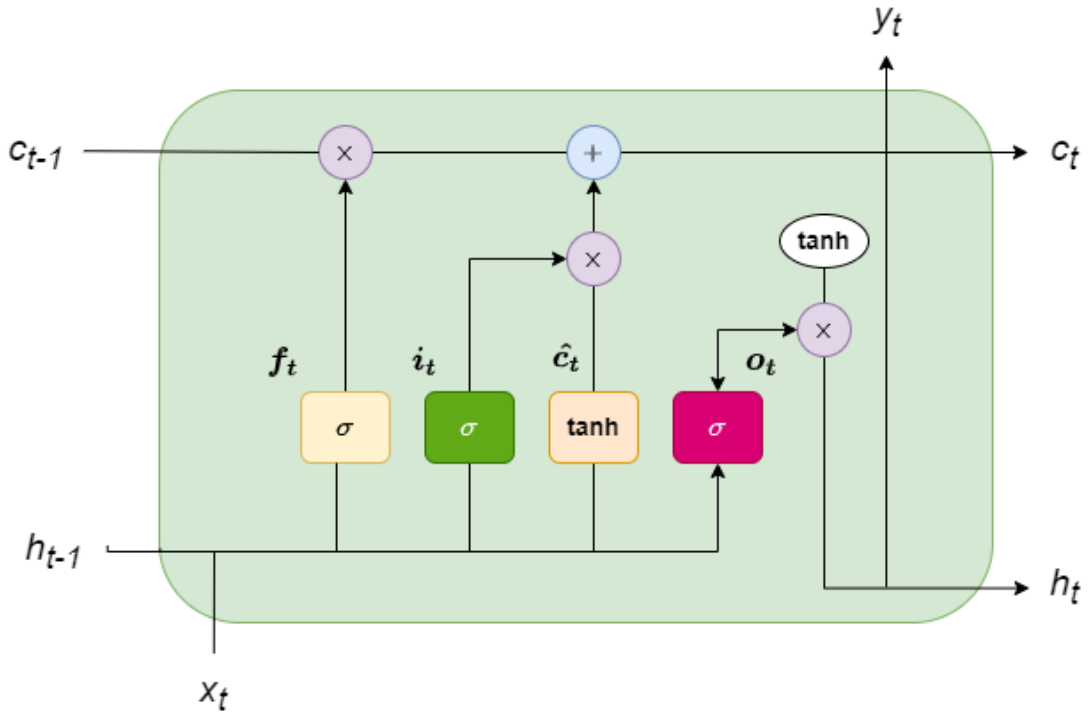

Figure 2　　LSTM 架構圖

通常之下，LSTM 會有一個輸入閥 $\boldsymbol{i}$、輸出閥 $\boldsymbol{o}$、遺忘閥 $\boldsymbol{f}$和記憶單元 $\boldsymbol{c}$，可以用以下式 1、式 2、式 3 來表示：

$$\begin{bmatrix} i_t \\ o_t \\ f_t \\ \tilde{c}_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \phi \end{bmatrix} \left(W_g^T \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_g \right) \quad \text{式 1}$$

$$c_t = c_{t-1} \odot f_t + \tilde{c}_t \odot i_t \quad \text{式 2}$$

$$h_t = o_t \odot \phi(c_t) \quad \text{式 3}$$

其中，在式 1 中i、o、f、$\tilde{c}$表示在時間點t下的輸入閥、輸出閥、遺忘閥以及記憶單元之更新值，W_g^T為權重值、b_g為偏移值，兩個皆是可訓練的參數值。而式 2 中的c_t則表示在時間點t下的記憶單元。函式$\sigma(\cdot)$和$\phi(\cdot)$分別是 Sigmoid 和 hyperbolic tangent，⊙代表元素對應乘積（Element-wise Product）。

雙向長短期記憶體 Bi－LSTM（Bi－directional LSTM）[20]由兩個方向不同的LSTM所組成，正向的（由左而右）LSTM，主要用來擷取序列過去資訊之特徵；而反向的（由右而左）主要用來擷取序列未來資訊之特徵。如此一來，假

設在時刻t時 ，記能夠使用$t-1$時刻之資訊，亦能夠使用$t+1$時刻之資訊。一般來說，Bi－LSTM 能夠同時利用到過去和未來的資訊，會比 LSTM 預測的結果來的更加準確。每個 Bi－LSTM 單元的更新可以寫成以下公式：

$$h_t = \vec{h}_t \oplus \overleftarrow{h}_t = \text{Bi-LSTM}(x_t, \vec{h}_{t-1}, \overleftarrow{h}_{t+1}, \theta) \quad \text{式 4}$$

其中，$\vec{h}_t$和$\overleftarrow{h}_t$分別代表是正向和反向 LSTM 在位置t時之隱藏狀態；$\oplus$代表是串連運算子；θ代表 Bi－LSTM 中的所有參數。

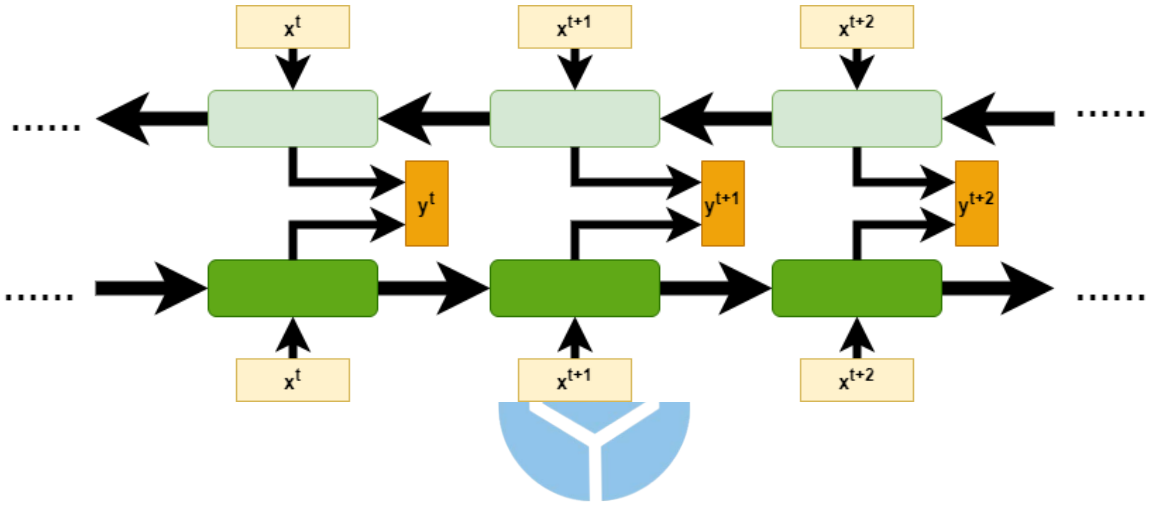

Figure 3　　Bi－LSTM 架構圖

2.1.3 Transformer

Transformer 是由 Google 在 2017 年所提出的架構[21]，在上一章節中可以得知RNN的特點在於每次的輸入都能保有之前得到的資訊，雖然經過改良的RNN，即 LSTM 已經能更有效率的保留長距離的資訊，但因其每次計算都得依賴前一次之計算結果，無法進行平行的運算，導致訓練時需耗費非常長的時間，且倘若時序拉的越長 LSTM 也會丟失更多的久遠的資訊[21] 。因此，Transformer 的架構就誕生了。

Transformer 不使用任何卷積神經網路和遞迴神經網路，而是使用了自注意力機制（Self-attention），這個機制有兩大特點：一為它可以平行化的運算，二

則是它每一個輸出的向量都看過每一個輸入的序列。重要的是，可以把所有 RNN 做的到的事都換成以 Self-Attention 來達到。如：Transformer 的 Self-Attention 從根本上解決了長距離記憶丟失問題。Self-Attention 會以遍歷的方式計算序列中任意兩個詞之間的相關性，所以不管兩個詞相隔多遠，都能捕捉到之間的依賴關係，從根本上解決難以建立長時依賴的問題。

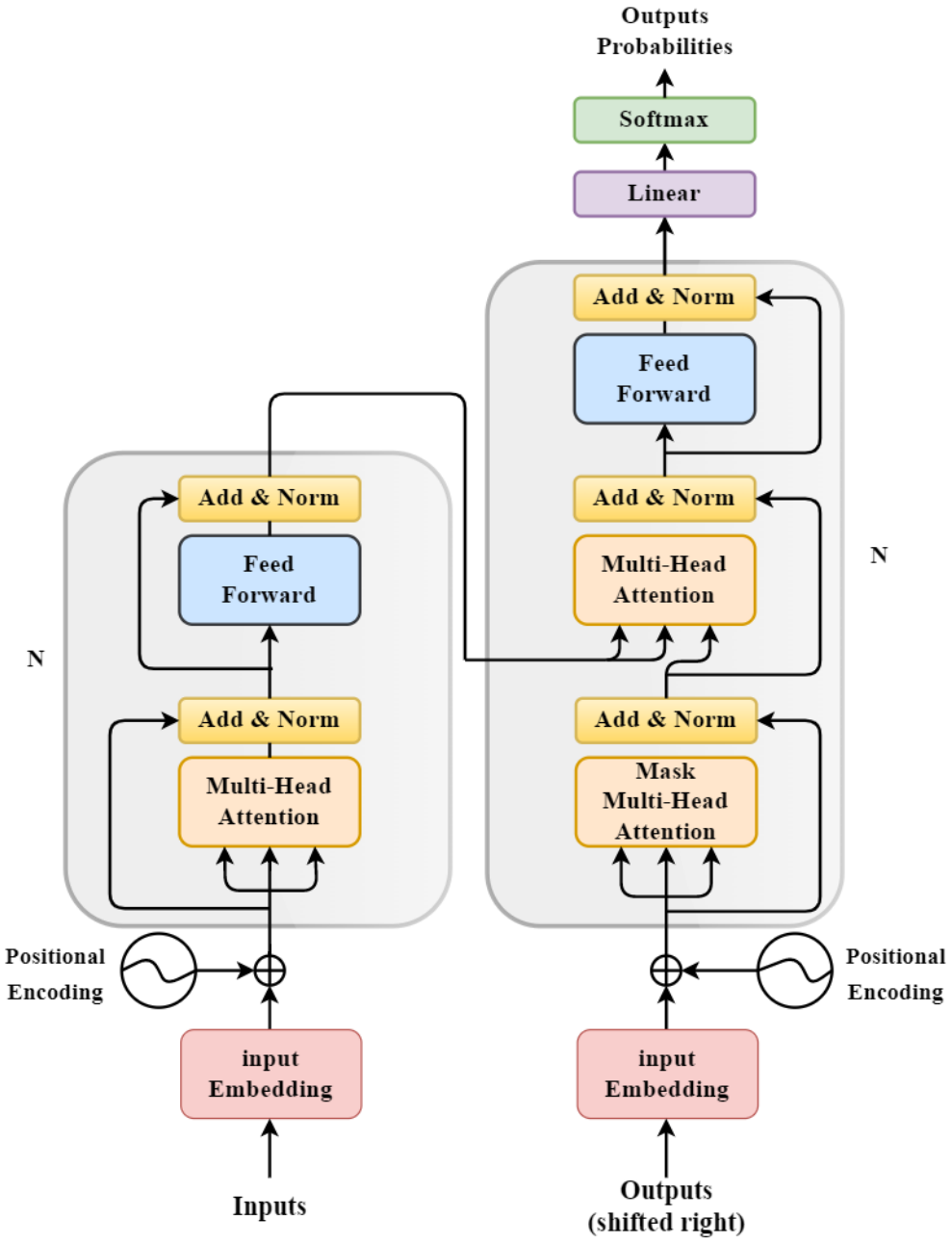

Figure 4　　Transformer 架構

Transformer 是採用一個 Seq2Seq 架構所組成，如 Figure 4 所示，左方為編碼器（Encoder），右方為解碼器（Decoder）。首先，輸入的序列由左邊編碼器部分開始先經過嵌入層獲得字符向量（Token Embedding），以及經由位置向量（Positional Embedding）來記錄各單詞在語句之間的位置或次序關係，而兩向量的大小一致。詞向量採用隨機初始化的方式在訓練時進行修正；而位置向量則是可以採用訓練或固定的方式得到。而論文中依據式 5、式 6 來獲得位置向量：

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad \text{式 5}$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad \text{式 6}$$

其中pos表示詞在序列中的位置，$2i$表示位置向量內的偶數維度，$2i+1$表示位置向量內的奇數維度，d_{model}則代表模型的維度（與詞向量相同維度）。

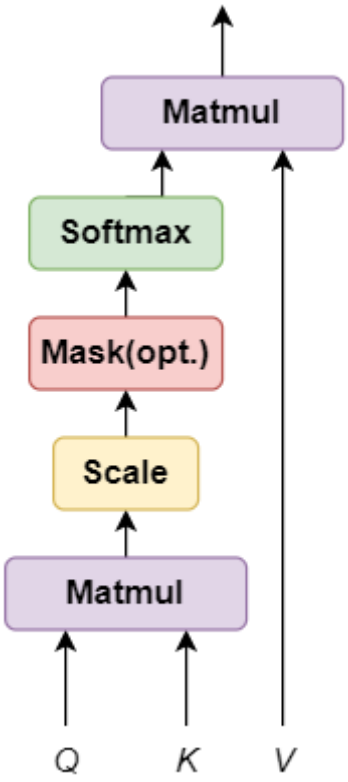

Figure 5　　Scale Dot-Product Attention

編碼器主要由多頭自注意力機制（Multi-Head Attention）和全連接層

（Fully Connected Feed-Forward Network）所組成，其中多頭注意力模塊中包含多個 Scale Dot-Product Attention 組成。同時架構中的子層也使用殘差網路（Residual Connection）及正規化（Normalization）來避免過擬合（Overfitting）問題。

Scale Dot-Product Attention 的三個輸入向量 Q（Query）、K（Key）、V（Value）起始相同，經由相對應的權重W^q、W^k以及W^v進行線性轉換後，將 Q 對所有的 K 做點積運算後除以$\sqrt{d_k}$進行正規化後經過 Softmax 再與 V 點積的方式計算注意力分數，如式 7 所示：

$$\text{Attention}(Q,K,V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V \qquad \text{式 7}$$

其中，d_k為 K 之維度。

Q、K、V所做的事情類似於資訊檢索中，使用者輸入檢索詞彙 Query，而搜尋引擎根據 Query 為你匹配 Key，最後根據 Query 和 Key 的相關性去找到合適的內容 Value。而在 Transformer 中，Query 與 Key 點積就可以獲得當前樣本關係向量，這樣每一個元素可以看做是當前樣本與序列中其他樣本之間的關係向量。獲得樣本之間的關係後，只需要將歸一化後乘以 Value 矩陣，最終加權輸出。Value 中的每一行是序列的一個樣本，其中每一維輸出，相當於是所有輸入序列樣本對應維度的加權和。

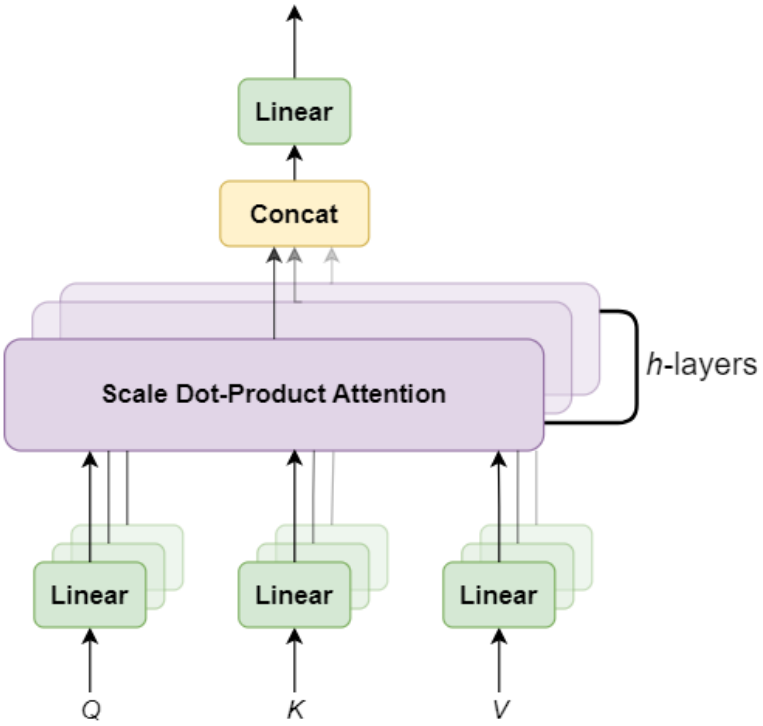

Figure 6 Multi-Head Attention

而 Multi-Head Attention 是對 Q、K、V 做h次線性映射後，再將最後結果串接起來，如 Figure 6 架構，以尋找序列的上下文關係，如式 8、式 9 所示。

$$\text{MultiHead}(Q,K,V) = \text{Concat}(head_i,\dots,head_h)W^O \quad \text{式 8}$$

$$\text{where head}_i = \text{Attention}_i(Q,K,V) \quad \text{式 9}$$

最後的全連接層則是將序列x經過兩個線性轉換，並且用 ReLU 激活函數，如式 10 所示：

$$\text{FFN}(x) = \max(0, xW_1 + b_1)\,W_2 + b_2 \quad \text{式 10}$$

解碼器的部分大多和編碼器架構一樣，第一個差別在於解碼器多了一層 Masked Multi-Head Attention，目的在於為了防止解碼器關注當前位置之未來的資訊，因此需要將未來資訊進行遮掩。第二個不同在於 Multi-Head Attention 的輸入 Q 來自編碼器的，而 K、V 來自解碼器自己的輸出，此舉是為了讓解碼器能獲得先前的資訊。

2.2 預訓練詞向量

2.2.1 詞向量表示法

所謂的詞向量（Word Embedding）[22]，就是將單詞向量化，將文章中的單詞用向量來表示。詞向量表示法的學習來自 2003 年被提出的神經網路語言模型[23]，其使用前饋神經網路（Feed-forward Neural Network）來建立一個 N 連語言模型（*N*-gram Language Model），在這個模型下自然地獲得了每一個詞的詞向量。詞向量是在自然語言處理中被廣泛使用的方法，將文章中的詞語轉變為特定的向量後，即可將其應用於各種機器學習的算法之中。一般來說，詞向量主要有兩種形式，分別為稀疏向量和密集向量。

稀疏向量又稱作 One-hot Representation，簡單來說就是使用一個很長的向量來表示一個詞，而長度取決於辭典大小N，在向量中只會有一個位置為 1，其餘部分皆為 0，1 所在位置會對應到這個詞在辭典中的索引。舉例來說，假如有一個辭典為〔蔡英文，韓國瑜，郭台銘〕，則「蔡英文」對應的詞向量就為[1,0,0]，而「韓國瑜」所對應的詞向量為[0,1,0]。One-hot Representation 的表示法之優點是不需要經過繁瑣的計算，但缺點是如果辭典過大，可能會造成詞向量維度過大的問題，並且此種方法無法從詞向量去看出兩詞之間的有何關係，因此在目前的研究上較少使用。

密集向量又稱為 Distributed Representation[22]，及分布式向量表示法，此種方法可以克服 One-hot Representation 可能造成維度過大之缺點，其基本思路是透過訓練將每個詞對應成一個固定長度的短向量，所有這些向量就可以構成一個詞向量空間，每一個向量可是為該空間上的一個點。此時向量的長度可以自由選擇，不必依賴於辭典規模大小。這種表是方法更精確的表現出近義詞之間的關係，比之 One-hot Representation 具有優勢。

目前常見的方法有連續詞袋模型（Continuous Bag-of-words，CBOW）[22]、

略字模型（Skip-gram）[22]、全局向量（Global Vectors for Word Representation，GloVe）[24]以及快文向量模型（FastText）[25]。

詞向量模型是考慮詞語位置關係的一種模型。通過大量語料的訓練，將每一個詞語映射到高維度的向量當中，通過求餘弦的方式，可以判斷兩個詞語之間的關係，例如 John 和 Mary 在詞向量模型中，他們的餘弦值可能就接近 1，因為這兩個都是人名，Taipei 和 John 的餘弦值可能就接近 0，因為一個是人名一個是地名。現在常用 Word2Vec 構成詞向量模型，它的底層採用基於 CBOW 和 Skip-Gram 算法的神經網絡模型。

2.2.2 連續詞袋模型和跳字模型

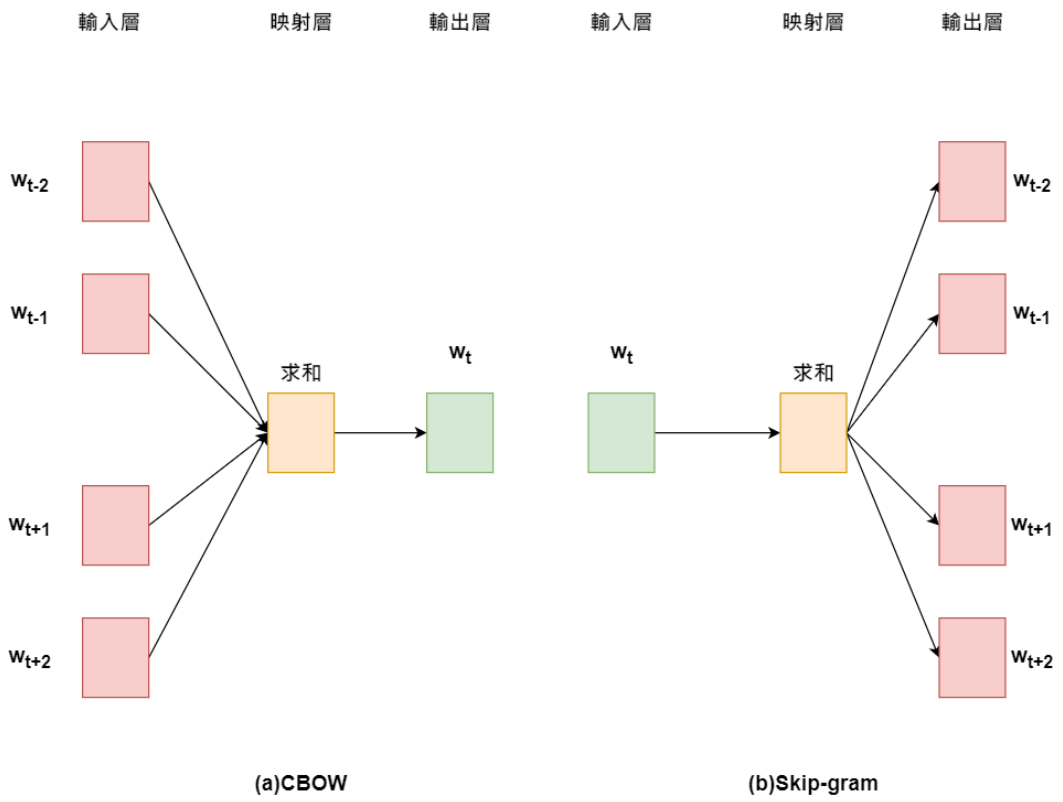

Figure 7　　CBOW 模型與 Skip-gram 模型

連續詞袋模型和跳字模型的基本概念類似，其運用了神經網路的概念，是利用一個單詞與其上下文組成簡單的神經網路。連續詞袋模型是假設基於某中心詞在文本序列前後的背景詞來生成該中心詞。舉例來說，當給定長度為T的單詞序列$w_1, w_2, \dots, w_t, \dots, w_T$，連續詞袋模型其目標函數（Objective Function）為：

$$min\, L = -\sum_{t=1}^{T} log\, P\left(w_t | w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}\right) \quad \text{式 11}$$

其目的為最大化上下文$w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}$出現目標單詞$w_t$的機率，其中$c$表示詞$w_t$的上下文窗口大小（Window Size）。而條件機率$P(w_t | w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c})$則為：

$$P(w_t | w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}) = \frac{exp\left(\overrightarrow{w_t}^T \cdot \overline{w_t}\right)}{\sum_{k=1}^{|V|} exp\left(\overrightarrow{w_k}^T \cdot \overline{w_t}\right)} \quad \text{式 12}$$

其中，$|V|$辭典總詞數，$\overrightarrow{w_t}$為詞w_t的詞向量，$\overline{w_t}$為出現在單詞w_t的上下文單詞$w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}$的詞向量加權平均值。連續詞袋模型主要採用上下文預測目標詞的方式，其物理上的意義表示語意相近的詞語通常與其左右相鄰的詞差異不大。

而另一種跳字模型則與上述連續詞袋模型使用相反的目標函數來獲取詞向量，也就是採用目標單詞預測上下文，其目標函數為：

$$min\, L = -\sum_{t=1}^{T} log\, P\left(w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c} | w_t\right) \quad \text{式 13}$$

其目的為最大化目標單詞w_t出現上下文$w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c}$的機率，其中，$c$為目標單詞$w_t$的滑動窗口大小(Window size)。

跳字模型中，假設已知目標詞w_t，因所有輸出字皆完全獨立，因此可表示為：

$$P(w_{t-c}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+c} | w_t) = \sum_{t=1}^{T} \sum_{j=-c, j\neq 0}^{c} \log P\left(w_{t+j} | w_t\right) \quad \text{式 14}$$

其條件機率可表示為：

$$P(w_{t+j}|w_t) = \frac{exp(\vec{w}_{t+j} \cdot \vec{w}_t)}{\sum_{k=1}^{|V|} exp(\vec{w}_k \cdot \vec{w}_t)} \quad \text{式 15}$$

其中，$|V|$辭典總詞數，$\vec{w}_{t+j}$與$\vec{w}_t$分別為w_{t+j}與w_t之詞向量。由上述公式中可以發現整體的計算量和$|V|$辭典總詞數有著很大的關聯，因此後有 Markov 等人提出負採樣法（Negative Sampling Algorithm）[26]來加速整體的訓練過程。

2.2.3 全局向量表示法

全局向量表示法是一種基於全局詞頻統計的詞向量表示法。前一小節提到的 CBOW 模型與 Skip-gram 模型皆有著明顯的缺點就是只能利用固定長度的滑動窗口大小，即只考慮到短距離的詞彙關係，因此就有了 GloVe 這種可以考慮到資料集中所有詞彙間的相互關係的表示法誕生。

GloVe 訓練法注重全文資訊，以訓練文件中兩個文字共同出現的矩陣，語意相近的詞通常共現次數（Co-occur）較多。式 16 中，X_{ij}為單詞w_j出現在單詞w_i上下文中的次數，X_i為單詞w_i作為其他單詞上下文的次數。

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} \quad \text{式 16}$$

模型的基本概念公式:：

$$F(\vec{w_i}, \vec{w_j}, \vec{w_k}) = \frac{P_{ik}}{P_{jk}} \quad \text{式 17}$$

其中，$\vec{w_i}$表示單詞w_i的詞向量，$\vec{w_k}$表示單詞w_k的上下文詞向量（Context Vector），詞向量與上下文向量皆由模型訓練所得。模型最終用於訓練的損失函數為：

$$J = \sum_{i,j=1}^{|V|} f(X_{ij}) \left(\vec{w_i}^T \vec{w_j} + b_i + b_j - log(X_{ij})\right)^2 \quad \text{式 18}$$

其中，$|V|$為單詞表單詞總數，f為權重函數，b_i為單詞w_i的偏移量（Bias）。

2.2.4 ELMo

上述表示法在本質上屬於靜態詞向量，所謂靜態詞向量是指當模型訓練好之後，同一個詞在不同上下文中擁有相同的詞向量，也就是說這個詞向量是固定住的 ，但在人類的語言裡，這樣的作法其實是不合理的，因為文章的文字中常常會有一詞多義的情況，例如：「蘋果」一詞可能代表品牌名稱或水果名稱，因此詞向量應該需要根據上下文擁有不同的詞向量。在2018 年 NAACL 上 Peter 提出的 ELMo（Embedding from Language Models）[27]就是廣為人知的動態詞向量表示法。

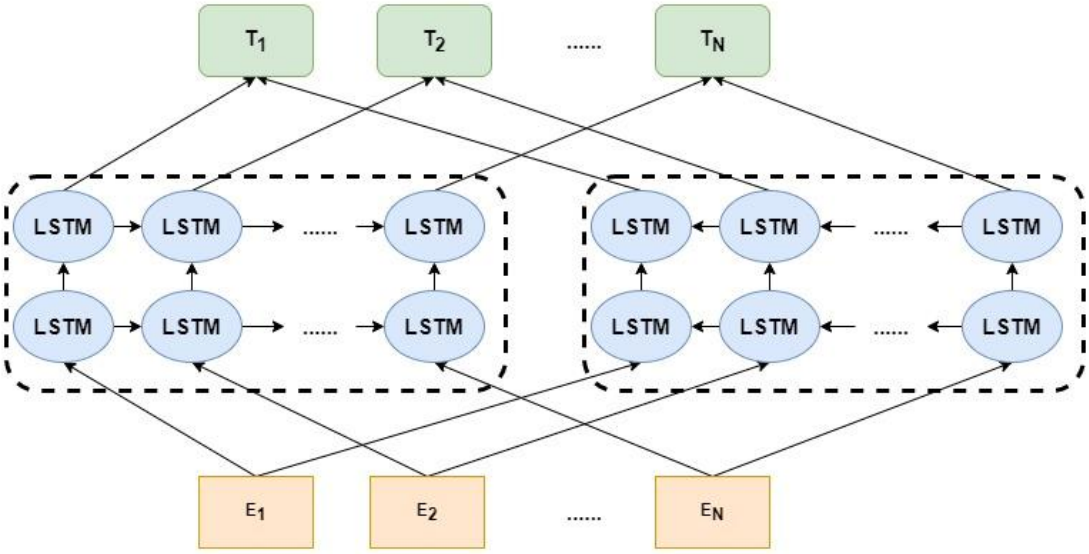

Figure 8　　ELMo 模型架構圖

ELMo 的核心思想主要體現於深度上下文（Deep Contextualized）上，與靜態詞向量不同的是，ELMo 除了提供臨時的詞向量之外，還提供生成這些詞向量的預訓練模型（Pre-trained Model），所以在使用此模型時已有了特定的詞向量，因此可以根據自己的資料集的實際上下文再去動態的微調（Fine-tune）詞向量以獲得符合上下文具體的含意。ELMo 主要利用雙向長短期記憶模型(BiLSTM)架構的語言模型（Language Model），學習依照字詞的上下文輸出適當的詞向量，其訓練目標可分為前向語言模型和後向語言

模型，如 Figure 8。算法如式 19 所示，前向語言模型是給定目標詞w_t之前的$w_1, w_2, \dots, w_{t-1}$預測目標詞w_t出現的機率；後向語言模型是給定目標詞w_t之後的$w_{t+1}, w_{t+2}, \dots, w_T$去預測目標詞$w_t$出現的機率，如式 20 所示：

$$P(w_1, w_2, \dots, w_T) = \prod_{t=1}^{T} P(w_t | w_1, w_2, \dots, w_{t-1}) \quad \text{式 19}$$

$$P(w_1, w_2, \dots, w_T) = \prod_{t=1}^{T} P(w_t | w_{t+1}, w_{t+2}, \dots, w_T) \quad \text{式 20}$$

其目標函數為最大化下列式子：

$$\sum_{t=1}^{T} \left(\log P(w_t | w_1, w_2, \dots, w_{t-1}; \theta_x, \vec{\theta}_{LSTM}, \theta_s) + \log P(w_t | w_{t+1}, w_{t+2}, \dots, w_T; \theta_x, \overleftarrow{\theta}_{LSTM}, \theta_s) \right) \quad \text{式 21}$$

其中θ_x、θ_s為詞向量層以及 Softmax 層參數，$\vec{\theta}_{LSTM}$、$\overleftarrow{\theta}_{LSTM}$分別為前後向長短期記憶網路模型之參數。

ELMo 對於每一個詞t_k輸出如以下式 22 所示：

$$R_k = \left\{ h_{t,j}^{LM} \middle| j = 0, \dots, L \right\} \quad \text{式 22}$$

其中$h_{t,j}^{LM} = [\overrightarrow{h_{t,J}^{LM}}; \overleftarrow{h_{t,J}^{LM}}]$表示每$j$層第$t$個輸出表示。最後，再透過式 23 對當前任務進行微調，拿出每一層的輸出分別加上一個可學習的權重來進行學習。

$$E(R_k; w, \gamma) = \gamma \sum_{j=0}^{L} s_j \, h_{t,j}^{LM} \quad \text{式 23}$$

其中，γ為縮放因子，依照任務作調整，s_j為歸一化指數函數。

2.2.5 OpenAI GPT

GPT 的全名為 Generative Pre-training[28]，2018 年由 Open AI 發表的預訓練模型。GPT 與 ELMo 相似皆使用兩階段的訓練模式，利用語言模型模型的預訓練，最後透過微調的方式解決下游任務的問題。GPT 的架構是使用單向的

Transformer 的解碼器取代原本廣泛使用的 LSTM 來提取特徵，也就是只根據當前目標詞的上文（Context-before）來預測該目標詞來進行無監督式的學習方式如 Figure 9。

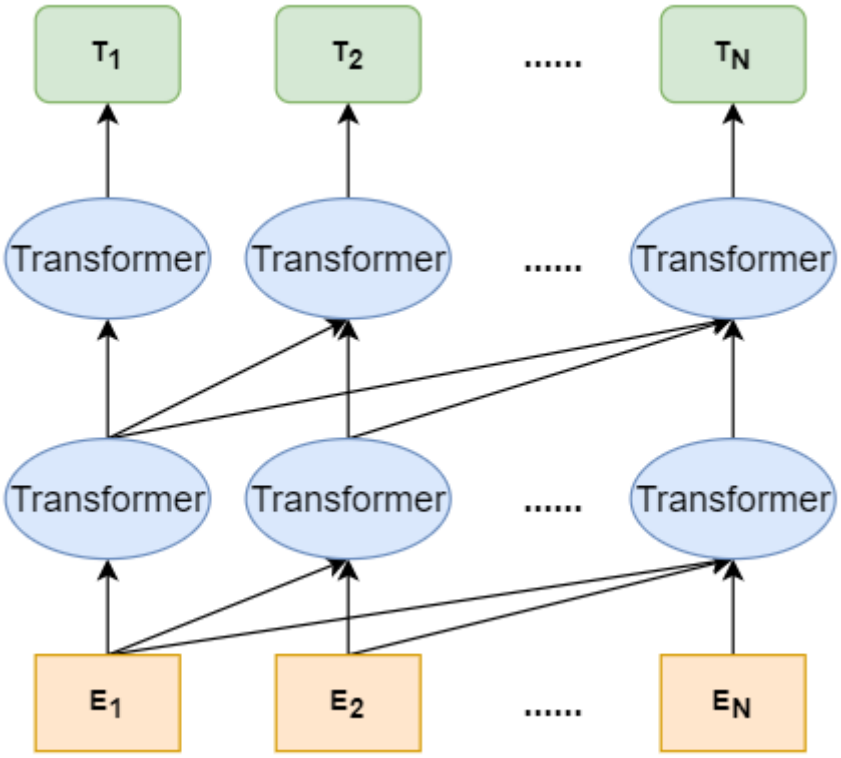

Figure 9　　OpenAI-GPT 模型架構圖

在第一階段語言模型的預訓練中，當給定長度為T的單詞序列$U = \{w_1, w_2, \dots, w_T\}$，其目標函數語標準語言模型之目標函數相同，如式 24 所示：

$$L(U) = \sum_i logP(w_i|w_{i-k}, \dots, w_{i-1}; \theta) \quad \text{式 24}$$

而 GPT 語言模型輸出如式 25、式 26、 式 27 所示：

$$h_0 = UW_e + W_p \quad \text{式 25}$$

$$h_l = transformer_{block(h_{l-1})} \forall l \in [1 \dots n] \quad \text{式 26}$$

$$P(u) = \text{Softmax}(h_n W_e^T) \quad \text{式 27}$$

其中，k表示考慮目標詞之前多少字，即滑動窗口大小，θ為 GPT 模型參數，

W_e和W_p表示詞向量矩陣以及位置資訊矩陣，n為 GPT 中 Transformer 之層數。

第二階段，只要在預訓練模型的最後加入一層全連接層與 Softmax 就能針對下游任務對整個 GPT 模型進行微調。

2.2.6 BERT

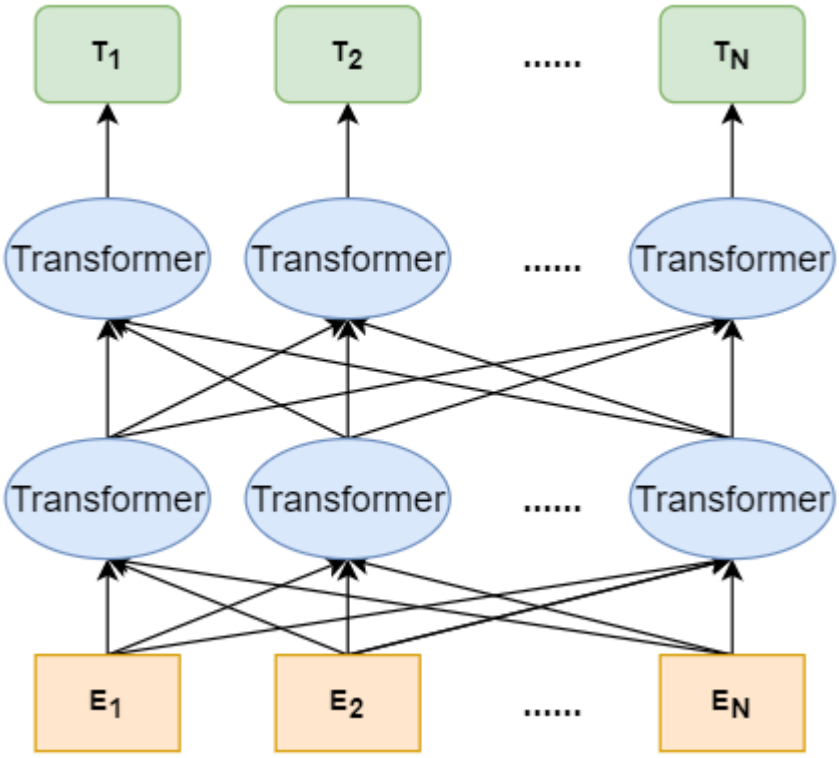

Figure 10　BERT 模型架構圖

BERT 全名為 Bidirectional Encoder Representation from Transformers[29]，是由 Google 在 2018 年發表的模型，這個模型廣為人知的模型橫掃了 NLP 的所有評估基準，為 NLP 開創了一個新的時代，甚至到現在很多 State Of The Art 的模型也都是 BERT 的思想延伸或衍生而來[30-36]。BERT 從名稱上就可以看得出來和上述的 GPT 不同，BERT 語言模型的架構是利用雙向的 Transformer 的編碼器部分，如 Figure 10。BERT 也是一個兩階段的訓練模型，第一階段經由 Google 預訓練在比 GPT 更龐大的資料上，再讓使用者再去對下游任務進行微調。BERT

在預訓練語言模型的階段新增了兩樣新穎的任務進行訓練，預訓練的第一個任務是遮掩語言模型（Masked Language Model，簡稱 MLM），此任務的做法是將輸入文字序列以 15%比例隨機遮掩單詞，並替換這些單詞成特殊符號[Mask]，讓模型根據[Mask]的上下文去預測[Mask]的正確單詞。舉例來說：

原本句子：今天天氣真冷，適合在家睡覺。

屏障句子：今天天氣真[Mask]，適合在[Mask]睡覺。

在預訓練時模型大量學習[Mask]的單詞，但在實際使用時並無[Mask]標記，這樣是明顯有問題的，所以 BERT 在 15%機率隨機單詞替換成[Mask]，又以其中 80%機率真的替換成[Mask]標記，10%維持原詞不變，最後 10%則隨機替換成其他單詞，藉此穩定訓練。

預訓練的第二個任務是下一句預測任務（Next Sentence Prediction），即是給予模型兩個句子，讓模型判斷第二句是否為第一句的後續句子。在此任務的訓練集中有一半是正確前後關係句子，而另一半則隨機挑選當作後續句子。BERT 的輸入型式需額外搭配[CLS]和[SEP]等特殊標記一起輸入，[CLS]代表句子的開頭，[SEP]則代表分句，舉例來說：

原始句子：今天天氣真冷，適合在家睡覺。

輸入型式：[CLS]今天天氣真冷[SEP]適合在家睡覺[SEP]

BERT 的預訓練中需訓練三種向量的表示法，除了字符向量（Token Embedding）和位置向量（Position Embedding）之外，還需額外訓練一個斷句向量（Segment Embedding）用來表示[SEP]的上下句關係。如圖所示 BERT 的輸入為此三種向量疊加，如 Figure 11 所示。[CLS]向量可以代表此句的句子向量，對於後續下游句子分類相關任務的幫助極大。

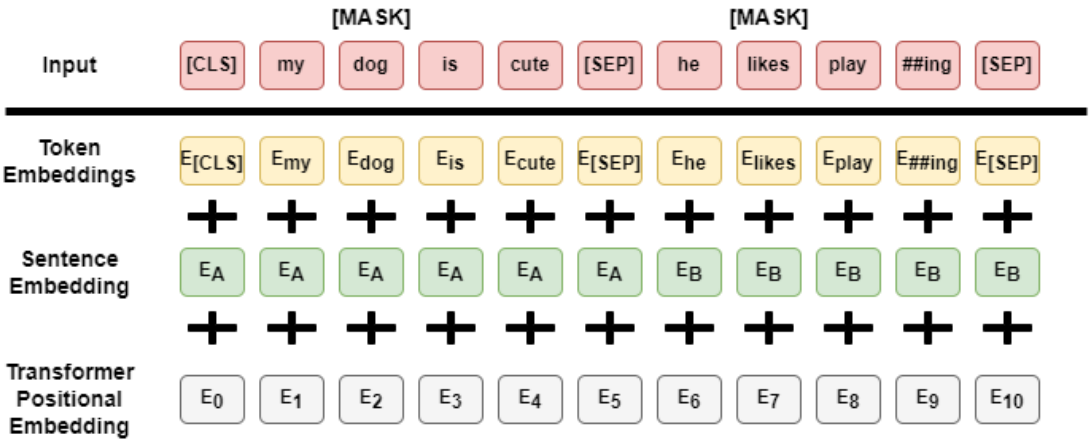

Figure 11　　BERT 輸入向量示意圖

2.3 監督式學習斷詞標註方式

2.3.1　四分類法

利用機器學習或深度學習的演算法在處理斷詞問題上，一般可以視為一種分類問題[37]，而目前最常見的方法就是字元分類。以中文來說，字元就代表一個字，如：「我」。詞則為多個字所合併，如：「我們」。字元分類也就是將一個詞中的依據每個字所在的位置給予對應的標籤，依照字元位置分為以下幾種標籤：詞的開始位置（Begin）、詞的中間（Middle）、詞的結尾（End）以及單一字元所組成的詞（Single）。

理論上中文字元可以存在詞的任何位置上，例如表格 2 所示，字元「中」可以出現在詞的開頭（B）、詞的中間（M）、詞的結尾（E）、以及單一字元的詞（S），所以 BMES 分類所要解決的問題就是每個字元該被歸類於哪個位置。假如現在輸入一篇句子，這個句子中的每個字元都會有位置的標籤，斷詞器就會依照這些標籤來切分出每個詞。例如：「今天天氣很好」，斷詞後的結果為「今天／天氣／很／好」，而其中每個字元的標記結果就會是「BEBESS」。

在 BMES 的分類問題中，因為一個字元可能會出現在不同的位置，因此一

旦標記錯誤，就會導致錯誤的斷詞結果。

表格 2　　字元「中」可以出現在詞的任意位置

B	中間
M	國民中學
E	集中
S	前進／中

2.3.2　六分類法

在 2006 年的 ACL 中，Chang-Ning Huang …等人提出的 Which Is Essential for Chinese Word Segmentation: Character versus Word [38]，將上述的方法加以改進成六種標籤的標記方式，依照字元位置可以分成以下幾種標籤：詞的開始位置（Begin）、詞的第二個位置（Begin1）、詞的第三個位置（Begin2）、詞的第四個位置至最後一個字前（Middle）、詞的結尾（End）。這裡假設斷詞器可以完整地對「不識廬山真面目」進行斷詞，得到的最後標記則是「BB_1B_2MMME」。在論文中有針對使用四分類法和六分類法甚至是五分類的的實驗，實驗顯示六分類法能略為提升模型效能，但目前在中文斷詞領域依舊以四分類法為主流。

2.4 本論文相關斷詞模型

2.4.1　長詞優先斷詞

長詞優先法（Maximum Matching Algorithm）是很普遍使用辭典比對斷詞方法，其中又分為正向長詞優先法（Forward Maximum Matching Algorithm）、反向長詞優先法（Backward Maximum Matching Algorithm）、以及雙向長詞優先法（Bi-directional Maximum Matching Algorithm）。首先，使用長詞優先法需要先準備一個辭典，其斷詞的方法是由句子的一端開始，試著比對出在辭典中最長

的詞，若符合辭典，則句子去除此詞後繼續向下依序比對，直到句子另一端結束。由此可知，長詞優先法的準確率取決於所用辭典的大小。

「正向長詞優先法」和「反向長詞優先法」的差別在於，正向長詞優先法從句子的開頭開始往句子的結尾掃描比對，而反向長詞優先法則是從句子結尾開始往句子開頭掃描比對，如表格 3 所示。

表格 3　　正向長詞優先法與反向長詞優先法比較

例句	正向長詞優先法	反向長詞優先法
高雄市民族藥局	高雄／市民／族／藥局	高雄市／民族／藥局
新崛江人潮壅擠	新崛江／人潮／壅擠	新崛江／人潮／壅擠

最後，「雙向長詞優先法」則是先以正向和反向長詞優先法斷詞後，根據其結果取兩種方法中成詞數較少的當作最後結果。以表格 3 例子來看，雙向長詞優先法最後的結果會以反向長詞優先法的結果為輸出。

此外，在本論文實驗中發現，長詞優先法在未知詞的處理上會有非常差的效果，因為未知詞就代表著在辭典裡並未出現過，所以無法比對斷詞，最後只能以單字詞當作最後結果。

2.4.2　監督式學習方法

2.4.2.1　隱藏式馬可夫模型

隱藏式馬可夫模型（Hidden Markov Model，HMM)[39, 40]是一個經典的機器學習模型，在語音識別、自然語言處理…等領域，都得到了廣泛的應用。隨著深度學習的崛起，如 RNN、LSTM 等神經序列模型的崛起後，HMM 的地位才有所下降。

HMM 主要透過觀測到的值去推測出目前狀態的演算法，其中狀態代表著隱藏狀態（Hidden State）。HMM 為一個生成式模型（Generative Model），所謂的

生成式模型為建立在聯合機率下的模型，也就是$P(X,Y)$。HMM 可以解決序列標註問題，在序列標註問題中，序列數據可以觀測到的部分稱為觀測序列；而從序列無法觀測到的則是稱為隱藏狀態序列；因此在斷詞中，輸入的句子為觀測序列，而斷詞的結果則為狀態序列。雖然無法得知目前的隱藏狀態，但可以透過觀測序列與狀態之間的轉移機率（Transition Probability）和發射機率（Emission Probability）進而推測出目前隱藏狀態，如 Figure 12。

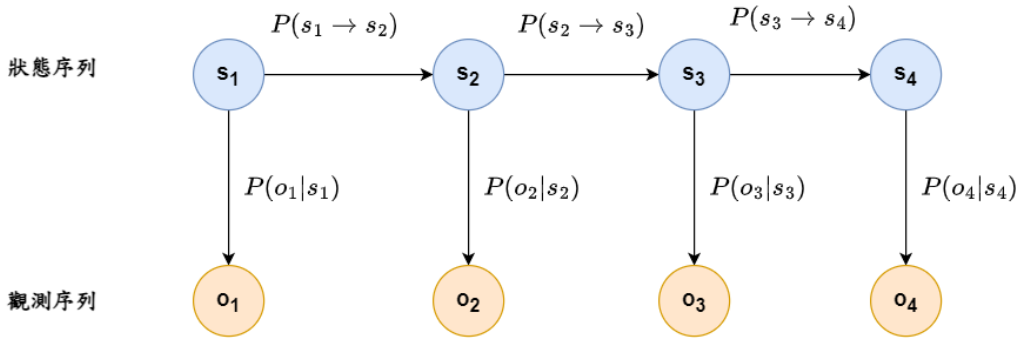

Figure 12　　HMM 節點依賴圖

首先，定義幾個關於 HMM 的參數以及說明 HMM 的兩個假設：

1. $\lambda = (\pi, A, B)$，λ為 HMM 之參數，其中包含π為初始機率（Initial Probability）、A為轉移機率、B為發射機率。
2. 觀測狀態集合$N = \{o_1, o_2, o_3, .., o_n\}$，此為輸入之句子的每一個字。
3. 隱藏狀態集合$M = \{s_1, s_2, s_3, .., s_m\}$，此為斷詞中 BMES 標註。
4. 狀態轉移矩陣T，紀錄兩個狀態間轉移之機率，$A = [a_{ij}]_{M*M}$，其中，$a_{ij} = P(s_{t+1}|s_t)$。
5. 狀態發射矩陣E，紀錄隱藏狀態到觀測狀態的機率，$B = [b_{ij}]_{N*M}$，其中$b_{ij} = P(o_t|s_t)$，o_t即第 t 個觀測節點，s_t即第 t 個隱狀態節點。
6. 一階馬可夫性假設（Markov Assumption）：假設當前狀態只與前一狀態有關。

7. 觀測獨立性假設：假設任意時刻的觀測值只依賴於當前時刻的馬可夫鏈的狀態（Markov Chain），與其他時刻的觀測值或狀態無關。

根據上面定義，接著說明 HMM 中三個基本問題：

一、機率計算問題：根據模型參數$\lambda = (\pi, A, B)$和觀測序列$N = \{o_1, o_2, o_3, .., o_n\}$，計算出此觀測序列之機率$P(N|\lambda)$，通常使用前向-後向演算法（Forward-Backward Algo.）來解決。

二、解碼問題：已知模型參數$\lambda = (\pi, A, B)$和觀測序列$N = \{o_1, o_2, o_3, .., o_n\}$，如何得到最大機率的隱藏狀態序列$M = \{s_1, s_2, s_3, .., s_m\}$，通常使用維特比演算法（Viterbi Algorithm）來進行解碼。

三、參數學習問題：根據觀測序列$N = \{o_1, o_2, o_3, .., o_n\}$，學習模型參數$\lambda = (\pi, A, B)$最大化$P(N|\lambda)$。

2.4.2.2 最大熵馬可夫模型

最大熵馬可夫模型（Maximum Entropy Markov Model，MEMM）[41]的思想是利用 HMM 的框架來預測給定觀測序列下的隱藏狀態序列，因此 MEMM 為一個判別式模型（Discriminative Model），所謂判別式模型就是建立在條件機率下的模型，即$P(Y|X)$。

在 HMM 的假設中，每個觀測序列中的值之間需要是互相獨立的，但在真實情況中觀測數據其實是有關係的，如：詞性、標點符號…等等。除此之外，上一章節說過 HMM 為生成式模型，但在序列標註的問題裡，要解決的問題是給定觀測序列下預測狀態序列，非常符合判別式模型的建立方式，因此使用 HMM 這類建立在聯合機率下的模型並不是這麼適合。和 HMM 不同的是，MEMM 當前狀態依賴於前一個隱藏狀態以及當前觀測值，如 Figure 13 和式 28 所示。

$$P(s_t|s_{t-1}, o_t), t = 1, \dots, n \qquad \text{式 28}$$

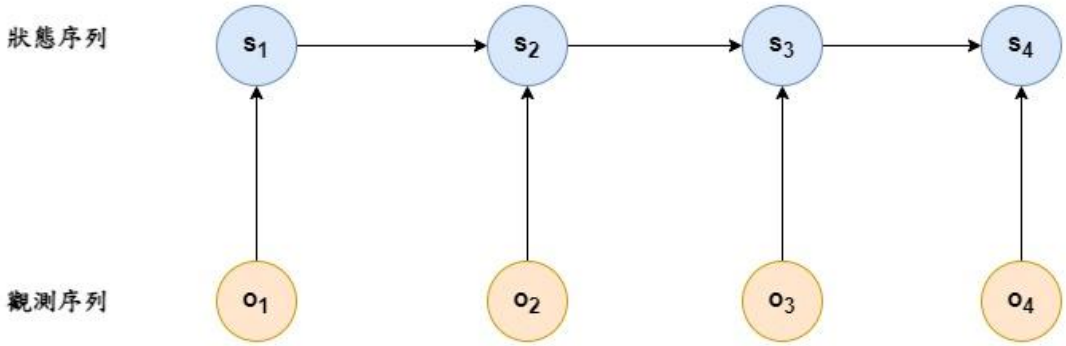

Figure 13　　MEMM 節點依賴圖

最大熵模型屬於對數線性模型，在給定訓練數據的條件下對模型進行最大概似估計或正規化最大概似估計：

$$P_w(y|x) = \frac{\exp\left(\sum_i w_i f_i(x,y)\right)}{Z_w(x)} \qquad \text{式 29}$$

其中 $Z_w(x) = \sum_y \exp\left(\sum_i w_i f_i(x,y)\right)$ 為歸一化因子，w 為最大熵模型參數，$f_i(x,y)$ 為特徵函數。最大熵模型並沒有假設特徵間相互獨立，允許定義特徵函數。

所以 MEMM 所需學習的條件機率套用入最大熵模型中，也就是結合式 28 和式 29 所得轉移狀態函數：

$$P(s|s',o) = \frac{\exp\left(\sum_\alpha \lambda_\alpha f_\alpha(o,s)\right)}{Z(o,s')} \qquad \text{式 30}$$

其中，λ_α 為特徵函數的權重，也是模型需要學習的參數。

對於整個序列來看的話，則 MEMM 可以如式 31 表示：

$$P(S|O) = \prod_{i=1}^{n} \frac{\exp\left(\sum_\alpha \lambda_\alpha f_\alpha(o,s)\right)}{Z(o,s_{t-1})}, t = 1, \ldots n \qquad \text{式 31}$$

MEMM 雖然克服了 HMM 的問題，但其本身也有一個令人詬病的問題存在，也就是標註偏移的問題，以 Figure 14 為例：

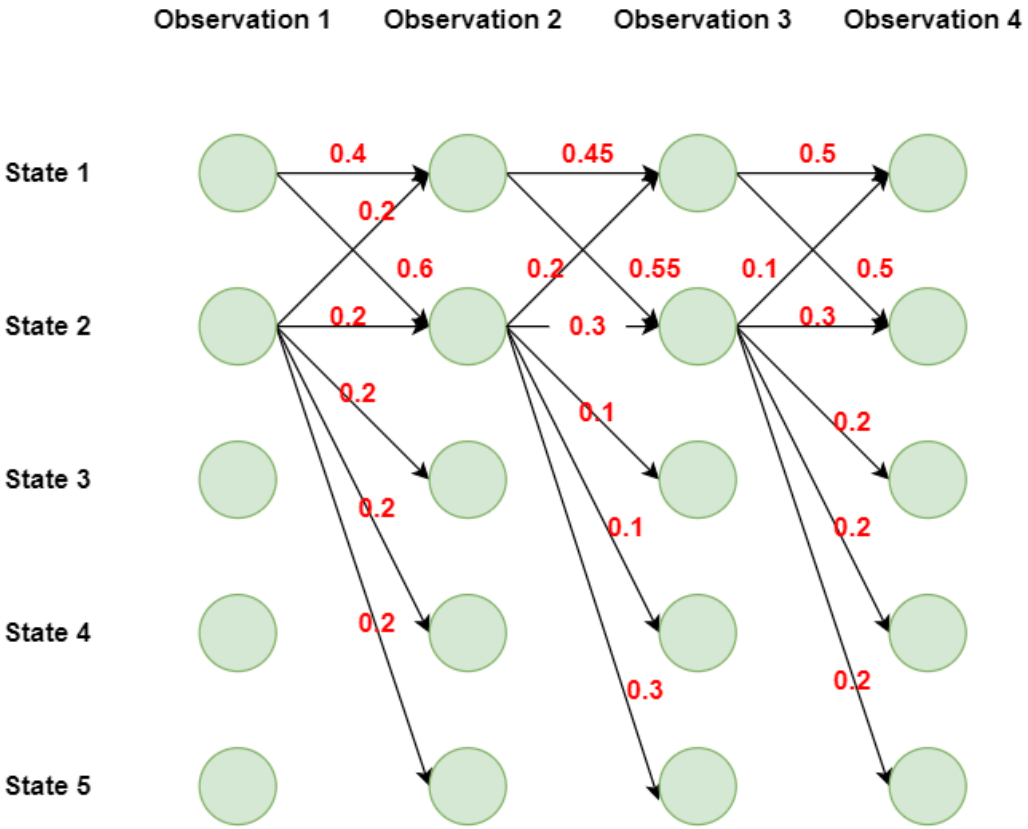

Figure 14　　標註偏移問題範例

使用維特比算法來解碼 MEMM，首先計算各路徑的機率值：

- 路徑一（$1 \to 1 \to 1 \to 1$）：$0.4 \times 0.45 \times 0.5 = 0.09$
- 路徑二（$2 \to 2 \to 2 \to 2$）：$0.2 \times 0.2 \times 0.3 = 0.018$
- 路徑三（$1 \to 2 \to 1 \to 2$）：$0.6 \times 0.2 \times 0.5 = 0.06$
- 路徑四（$1 \to 1 \to 2 \to 2$）：$0.4 \times 0.55 \times 0.3 = 0.066$

因此，最後結果可以得到路徑一為最優路徑，但從全局的角度來看：

- 無論觀測值，狀態 1 都傾向於轉移到狀態 2。
- 無論觀測值，狀態 2 都傾向於轉移到狀態 2。

MEMM 會造成這樣的原因是式 31 中，歸一化放在了指數的內部，視為局部歸一化，因此擁有更少轉移狀態的轉移機率普遍偏高，最後就會導致機率最大的路徑更容易轉移少的狀態。

2.4.2.3 條件式隨機域

條件式隨機域（Conditional Random Field，CRF）[42, 43]，由和 MEMM 大致相同的作者 Lafferty 等人於 2001 年提出，結合了最大熵馬可夫模型和隱藏式馬可夫模型（HMM）的特點，是一種無向圖模型或一種馬可夫隨機域，如 Figure 15。CRF 以隨機變量X序列為條件來描述狀態序列，以正式上的定義來說，可以定義$G = (V, E)$為一個無向圖，其中所包含的所有節點V為對應每個描述隨機變數S的元素s_t之隨機變數。如果每個隨機變數s_t遵守與 G 有關之馬可夫性質，(S, X) 便可視為一個條件式隨機域。

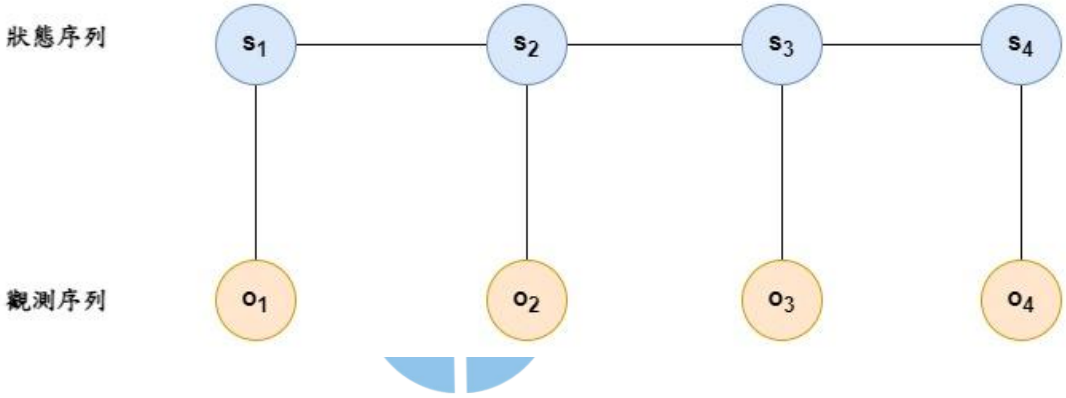

Figure 15　　CRF 節點依賴圖

在分詞、詞性標註和命名實體識別等序列標註任務中取得了很好的效果。而在這裡介紹的是用於序列標註問題上的線性鏈條件式隨機域（Linear Chain Conditional Random Field）。線性鏈條件式隨機域是由輸入序列來預測輸出序列的判別式機率模型，對於序列標註問題，則假設X是輸入代表待標註的觀測序列，Y是輸出代表標註的序列。CRF 和 MEMM 一樣屬於判別式的模型，因此 CRF 模型也是建立於$P(Y|X)$的條件機率下，且 CRF 也和 MEMM 一樣做了一階馬可夫假設，即當前狀態只與上一狀態有關，但是區別在於 CRF 的特徵採用了全域特徵，它把觀測序列當做整體來看所以它的特徵函數是全域的，它的特徵函數φ為：

$$\varphi(o_1 \dots o_m, s_1 \dots s_m) = \sum_{i=1}^{m} \phi(o_1 \dots o_m, i, s_{i-1}, s_i) \quad \text{式 32}$$

其中ϕ和 MEMM 的特徵函數是一致的，和 MEMM 的區別是，MEMM 的當前狀態與上一狀態和當前的輸入有關，而 CRF 當前的狀態與上一狀態和所有的輸入有關，這可以緩解標註偏移的問題。接下來的步驟和 MEMM 差不多了，只是特徵函數變為了φ，所以 CRF 的條件機率運算式為：

$$P(s_i|s_{i-1}, o_1 \dots o_m; w) = \frac{\exp\left(w \cdot \varphi(o_1 \dots o_m, i, s_{i-1}, s_i)\right)}{\sum_{s' \in S} \exp\left(w \cdot \varphi(o_1 \dots o_m, i, s_{i-1}, s')\right)} \quad \text{式 33}$$

而對於所有狀態來說可以寫成：

$$\begin{aligned} P(S|O) &= \prod_{i=1}^{m} P(s_i|s_{i-1}, o_1 \dots o_m) \\ &= \prod_{i=1}^{m} \frac{\exp\left(w \cdot \varphi(o_1 \dots o_m, i, s_{i-1}, s_i)\right)}{Z} \\ &= \frac{\exp\left(\sum_i w \cdot \varphi(o_1 \dots o_m, i, s_{i-1}, s_i)\right)}{Z} \\ &= \frac{\exp\left(\sum_i \sum_j w_j \cdot \phi_j(o_1 \dots o_m, i, s_{i-1}, s_i)\right)}{Z} \end{aligned} \quad \text{式 34}$$

其中，i表示當前所在位置，j表示第幾個特徵函數，$Z = \sum_{s' \in S} \exp\left(w \cdot \varphi(o_1 \dots o_m, i, s_{i-1}, s')\right)$。

2.4.2.4 雙向長短期記憶模型

此方法由 Google AI Language 在 EMNLP2018 上發表的 State-of-the-art Chinese word segmentation with Bi–LSTMs[44]，在 Transformer 架構問世以前，此模型在中文斷詞領域上達到 State Of The Art。此篇論文所提出的模型架構非常簡單，如 Figure 16 所示，將輸入的字序列轉換成字向量後通過雙向的 LSTM，最後透過 Softmax 輸出每個字最大機率的標記。其中，Figure 16 (a)中字向量是分別丟入前向 LSTM 和後向 LSTM 最後再合併一起進入最後一層全連接層進行 Softmax 輸出；而 Figure 16(b)則是字向量先經由後向 LSTM 後接著進入前向

LSTM 最後進入全連接層進行 Softmax 輸出。

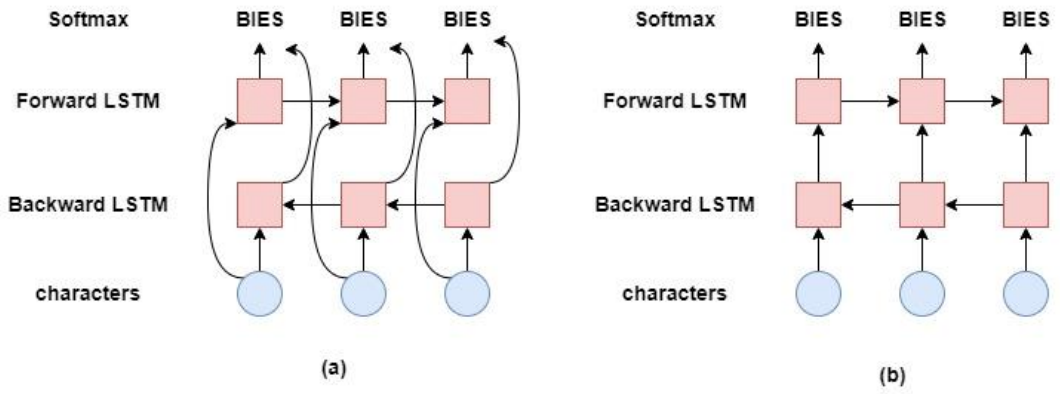

Figure 16　　Bi–LSTM 斷詞架構圖

從模型來看此方法非常的簡單，卻能達到當時中文斷詞的 State Of The Art，作者歸納出以下幾點原因：

1. 預訓練字向量（Pretrained Character Embedding），作者使用 Wang2Vec 作為預訓練的字向量，Wang2Vec 相較於 Word2Vec 多了字與字之間順序的訊息，並且結合了字與相鄰字之向量疊加輸入模型，舉例來說：輸入字序列為「今天天氣很好。」，當「今」這個字輸入模型時會一起疊加「今天」這個 Bigram 字向量一起輸入；而當「天」這個字輸入模型時會一起疊加「今天」和「天天」兩個 Bigram 字向量一起輸入。
2. 在 LSTM 的循環中加入 Dropout 機制針對上一個輸出的狀態抑制過度擬合。
3. 使用 Momentum-based Averaged SGD（Weiss et al.2015）[45]優化器訓練模型。

論文實驗指出以上三點的使用對模型提升了 0.3-0.8 不等的準確率，其中又以預訓練字向量提升最多。另外，論文中也顯示相對於以多個任務共享參數，針對資料集進行調整參數更能保證效能的提升。最後，在作者實驗中發現在有預訓練字向量的情況下，未登入詞（Out of Vocabulary，OOV）可以提升約 10%

的召回率（Recall），因此作者除了模型上的改進外，如何將外部辭典與模型結合應用也是很重要的一部分；另外，資料集的人工標記不一致也是造成效能無法提升的重要原因之一。

2.4.2.5 堆疊卷積神經網路結合條件式隨機域

2017 年由 Chunqi Wang…等人提出 Convolutional neural network with word embeddings for Chinese word segmentation[46]，使用帶有詞向量的卷積神經網路堆疊的方式來達到長距離的訊息捕捉，並且在最後的效能上達到和上述 LSTM 方法相近的成果。論文中作者認為在此之前很多模型在中文斷詞領域上已經能達到不錯的效果，但明顯有兩個問題：

1. 嚴重依賴手動設計的 Bi-gram 特徵，模型不擅長自動捕獲 *N*-gram 特徵。
2. 模型沒有使用完整的詞彙訊息。

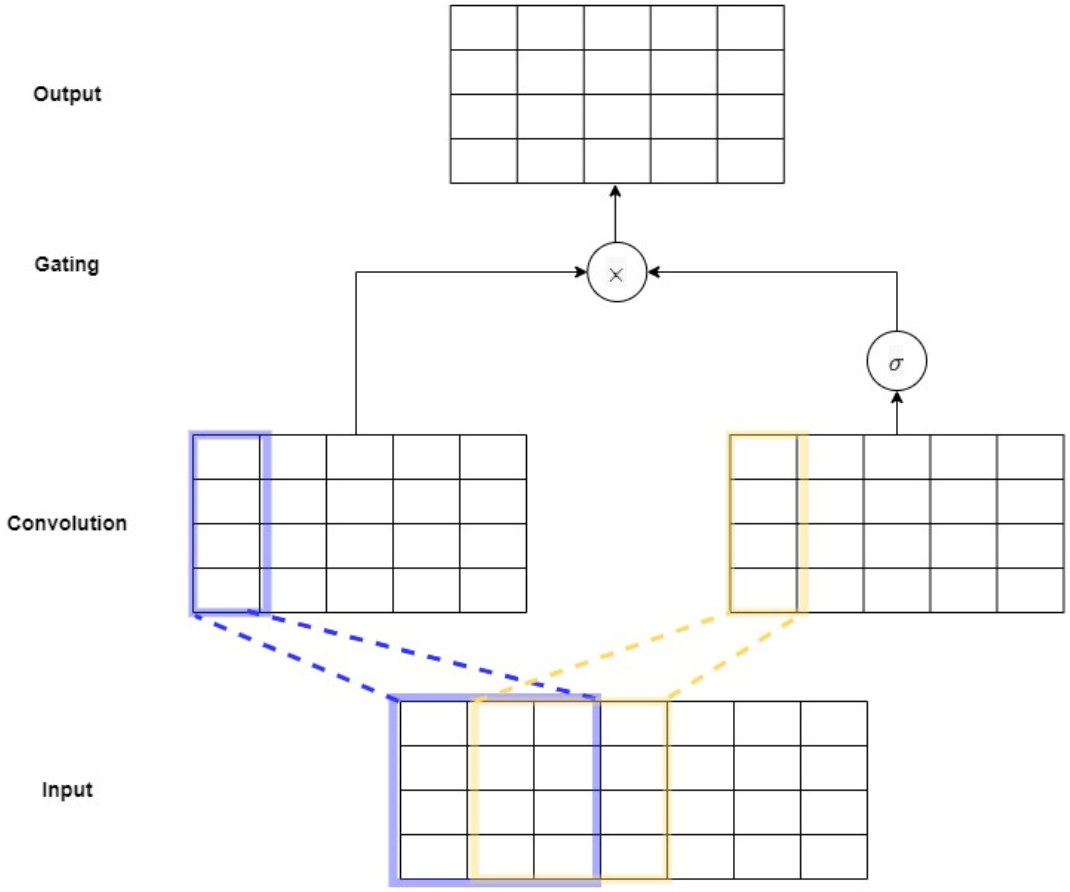

Figure 17　　Stacked – CNN-CRF 模型架構圖

該篇論文的模型卷積層架構大致如 Figure 17 所示，首先，將輸入字串序列轉換成預訓練過的字向量，並經過由一維的卷積核為 3 且步幅為 1 的卷積層掃描，舉例來說：若輸入字串序列為「今天天氣很好」且字向量維度為200，這裡的卷積核會先掃描「今天天」此三個字的3×200的矩陣，捕捉這三個字之間的關係，而步幅為 1 則代表卷積核會接著掃描「天天氣」這三個字的3×200矩陣…以此類推至字串結束。在此作者將經過卷積層後得到的特徵圖（Feature Map）額外複製一份，並將兩份特徵圖經過激勵函數（Activation Function）Gated Linear Unit[47]後將可得到輸出，以上操作該篇論文共做了 5 次，即五層閥門卷積層（Gated Convolution Layers）。實驗將輸入通道（Channel）的數量定義為N，輸出通道的數量為M，輸入長度為L，卷積核為 k，閥門卷積層數學如式 35 所示：

$$F(X) = (X * W + b) \otimes \sigma(X * V + c) \quad \text{式 35}$$

上式中*代表 1 維的卷積運算，X為維度$L \times N$的此層輸入，W、b、V、c皆為模型學習參數，$\otimes \sigma(\,)$代表 Gated Linear Unit，其中，σ代表 Sigmoid 函數而$\otimes$代表元素對應乘積（Element-wise Product）。

接著，作者使用 CRF 層當作模型的最後一層輸出，上述方法有介紹過 CRF 可以考慮到相鄰標籤的關係，因此作者不使用一般全連接層 Softmax 輸出每種標註的機率改而使用 CRF 層。模型的損失函數定義如式 36：

$$L(S, y^*) = -logP(y^*|S) \qquad \text{式 36}$$

其中，S為字串序列$(c_1, c_2, \ldots, c_L,)$，y^*為真實標註序列。在訓練過程中，透過反向傳播使損失函數最小；在測試過程中，則是使用維特比算法（Viterbi Algorithm）以最大機率找到標註序列。

作者認為基於字（Character-based）的中文斷詞擁有很大的靈活性及有效性，但卻無法好好利用對於詞的訊息。因此作者們提出了一個方法能使基於字的模型能有效利用到詞的資訊。此方法可以帶來兩種效益：

1. 能夠完整利用到詞資訊。
2. 大量的未標註資料能更有效的利用。

為了使用詞向量，作者設計了一系列的單詞特徵，如下表格 4 所示：

表格 4　　字結合詞特徵關係圖

Length	Feature
1	c_i
2	$c_{i-1}c_i$ 、 c_ic_{i+1}
3	$c_{i-2}c_{i-1}c_i$ 、 $c_{i-1}c_ic_{i+1}$ 、 $c_ic_{i+1}c_{i+2}$
4	$c_{i-3}c_{i-2}c_{i-1}c_i$ 、 $c_{i-2}c_{i-1}c_ic_{i+1}$ $c_{i-1}c_ic_{i+1}c_{i+2}$ 、 $c_ic_{i+1}c_{i+2}c_{i+3}$

其作法為給定句子$S = (c_1, c_2, \ldots, c_L,)$，在位置 i 的單詞特徵，只有包含c_i的單詞才可以視為單詞特徵。此論文限制了單詞最大長度為4，因為在中文的詞語

中，很少有超過 4 個字以上的詞。以上c_i的向量表示法$R(c_i)$如下式 37 所示：

$$R(c_i) = M_{char}[c_i] \oplus M_{word}[c_i] \oplus M_{char}[c_{i-1}c_i] \oplus \dots \oplus M_{char}[c_i c_{i+1} c_{i+2} c_{i+3}] \quad \text{式 37}$$

其中，$\oplus$表示串接的運算。具體使用方式如下：

1. 訓練一個不依賴單詞特徵的「教師」模型。
2. 利用教師模型對未標註數據D進行斷詞，並獲得自動斷詞數據D'。
3. 從D'構建辭典V_{word}。將V_{word}中未出現的所有單詞替換為 UNK（Unknown Word）。
4. 使用 Word2Vec 預訓練D'上的詞向量。
5. 使用預訓練好的詞向量和單詞特徵訓練「學生」模型。

此篇論文提供了參考如何將詞特徵整合至模型內使用，並且沒有使用任何外部標記的數據，詞向量只要來自大量自動斷詞的資料，結合詞特徵的方法也使得模型準確率得到大量的提升。

2.4.3 非監督式學習方法

2.4.3.1 基於分段語言模型的非監督斷詞法

傳統的非監督式中文斷詞大致可以分為判別式模型以及生成式模型，前者使用精心設計過的有效方法來進行候選詞分割，而後者則是著重於為中文設計統計模型，並找到生成機率最高最優分割。本篇論文為 EMNLP2018 所提出 Unsupervised neural word segmentation for chinese via segmental language modeling[48]，用於中文斷詞的分段語言模型（Segmental Language Model，簡稱 SLM）。在 SLM 中，使用的是類似於編碼器解碼器（Encoder-Decoder）的架構，上下文編碼器對先前的上下文進行編碼，而段解碼器則是遞增的生成每個段。此篇論文為第一個使用於中文斷詞的非監督神經網路模型，並且在 SIGHAN 2005 的四個不同資料集上達到能與 State Of The Art 的統計模型競爭的效能。

神經語言模型可以給出給定先前字元的下一個字元的條件機率，通常由遞迴神經網路（RNN）實現:

$$h_t = f(y_{t-1}, h_{t-1}) \quad \text{式 38}$$

$$p(y_t|y_{1:t-1}) = g(h_t, y_t) \quad \text{式 39}$$

其中，y_t是第t個字元的分布表示式，而h_t代表前一個字元的資訊。

SLM 和神經語言模型相似，因此 SLM 的模型目標為學習分段字元序列的聯合機率函數，因此對於每個分段，如式 40 所示：

$$\hat{p}\left(y_t^{(i)} \middle| y_{1:t-1}^{(i)}, y^{(1:i-1)}\right) = g(h_t^{(i)}, y_t^{(i)}) \quad \text{式 40}$$

其中，$y_t^{(i)}$是第i段中第t個字元的分布表示式，$y^{(1:i-1)}$是前面已分割的段。所有片段$y_{1:T_i}^{(i)}$的拼接正好是整個句子$y_{1:T}$，其中T_i是第i個片段$y^{(i)}$的長度，T是句子y的長度。

值得注意的地方是，雖然模型架構有上下文編碼器與分段解碼器，但 SLM 不是一個標準的 Encoder-Decoder 模型，因為解碼器所生成內容的和編碼器所提供的內容並不是一樣的。Figure 18 說明了 SLM 如何處理候選分割，SLM 使用候選分段y_1、$y_{2:3}$和y_4處理字串序列 $y = y_1y_2y_3y_4$，其中y_0是一個附加的開始符號，對所有句子都保持相同。

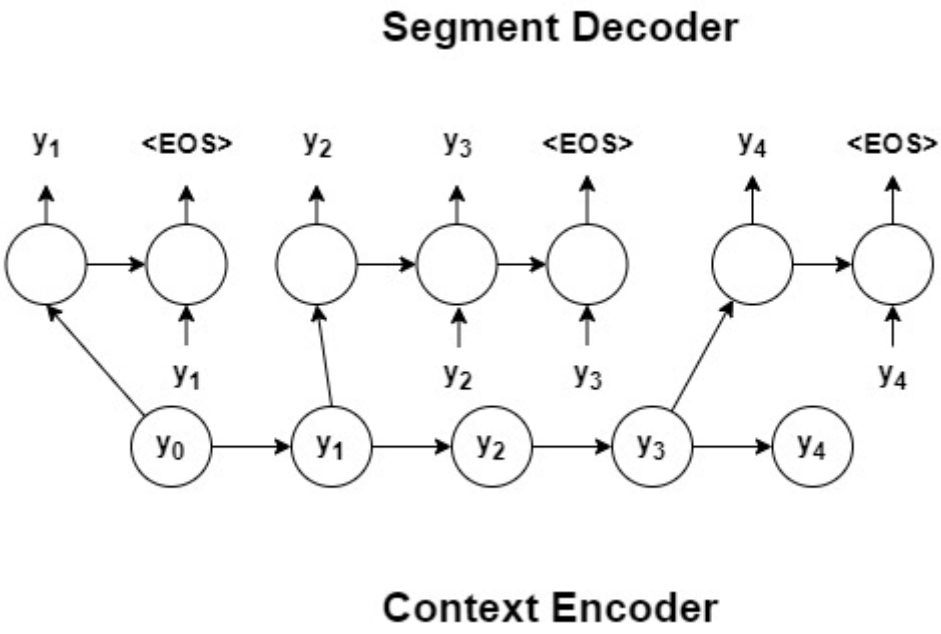

Figure 18　　SLM 具體斷詞流程

因為是非監督式的方法，所以給定的句子是沒有經過斷詞標註的，SLM 計算每個字的所有可能的斷詞序列的機率，如式 41 所示：

$$p(y_{1:T}) = \sum_{T_1,T_2,\ldots} \prod_i \hat{p}\left(y_{1:T_i}^{(i)}\right) = \sum_{T_1,T_2,\ldots} \prod_i \prod_{t=1}^{T_i+1} \hat{p}(y_t^{(i)} | y_{0:t-1}^{(i)}) \qquad \text{式 41}$$

其中$y_{T_{i+1}}^{(i)}$ =< eos >是每個段末尾段結束符號，$y_0^{(i)}$則表示上文$y_t^{(1:i-1)}$表示式。

對於句子生成，SLM 可以透過逐個生成片段並在生成結尾符號$< eos >$停止。時間複雜度與生成句子的長度成線性關係，因為模型可以保持上下文編碼器 LSTM 的隱藏狀態，並且在生成新單詞時來更新它。

最後，可以驗證 SLM 保留了語言模型的機率內容：

$$\sum_i P(s_i) = 1 \qquad \text{式 42}$$

其中，s_i列舉了所有可能的句子。

訓練的損失函數與語言模型類似，目標為最大化訓練資料集的對數概似估計（log-likelihood）：

$$L = -\log\, p(y_{1:T}) \qquad \text{式 43}$$

可以在給定$p(y_{1:0}) = 1$初始條件下，使用動態規劃（Dynamic Programming）在線性時間複雜度中計算損失目標函數。

$$p(y_{1:n}) = \sum_{k=1}^{K} p(y_{1:n-k})\hat{p}(y_{n-k+1:n}) \qquad \text{式 44}$$

其中，$p(\cdot)$是所有可能分割的聯合機率，$\hat{p}(\cdot)$是一個段的機率，K則是段的最大長度。

解碼時，作者使用動態規劃以類似於$\bar{p}(y_{1:0}) = 1$的方式找到線性時間中具有最大機率的分割，如式 45、式 46 所示：

$$\bar{p}(y_{1:\mathrm{n}}) = max_{k=1}^{K}\bar{p}(y_{1:\mathrm{n}-\mathrm{k}})\hat{p}(y_{n-k+1:n}) \qquad \text{式 45}$$

$$\delta(y_{1:\mathrm{n}}) = argmax_{k=1}^{K}\bar{p}(y_{1:\mathrm{n}-\mathrm{k}})\hat{p}(y_{n-k+1:n}) \qquad \text{式 46}$$

其中$\bar{p}$是最佳分割的機率，而δ用於解碼。

作者於該實驗中，將 SIGHAN 2005 資料集中的所有標點符號都替代為＜punc＞符號，英文替代為＜eng＞符號，阿拉伯數字則替代為＜num＞符號。模型以訓練集和測試集合併訓練，測試時只測試於測試集上。預訓練字向量使用 Word2Vec 預訓練於 Chinese Gigaword corpus 資料上。

另外，作者依據段的最大長度K分別訓練 3 個模型，即 SLM-2、SLM-3 及 SLM-4，但沒有訓練$K > 4$的模型，因為在中文裡很少有詞是大於四個字的。實驗結果顯示，在此三個模型中無法判斷出哪個模型的效能最好，因為 SIGHAN 2005 中所包含的 4 個資料集的標註情況並不一致。

2.4.3.2 使用雙向語言模型之非監督斷詞法

2021 年由 Lihao Wang…等人發表的 Unsupervised Word Segmentation with Bi-directional Neural Language Model [49]，在 SIGHAN 2005 的四個資料集中皆達到 State Of The Art 的效果。此篇論文架構與上述方法 SLM 架構相似，皆是採用類似於 Encoder-Decoder 的架構，編碼器一樣主要用於對輸入序列進行上下文資訊

的擷取，解碼器則是生成每個詞之最大機率。不過，在 SLM 中的編碼器是採用單向 LSTM 進行編碼，而此論文則是改用雙向 LSTM 的架構，更有利於捕捉上下文的長短距離的資訊。並且作者在解碼時提出了兩種演算法來組合雙向的上下文特徵，一種是通過將前向語言模型與後向語言模型所估計的斷詞機率的平均值來生成斷詞，另一種則是透過合併前向和後向語言模型兩模型分別產生的結果來生成斷詞。前者更有利於歧異性問題的解決，而後者則實現了整體更高的效能。在此將雙向語言斷詞模型代稱為 UCWS。

作者引入一個決定斷詞結果的潛在變數z。如果給定一個句子d，則z給定d和參數θ的機率可表示為下方式 47：

$$p_\theta(z|d) = \frac{p_\theta(d|z)p_\theta(z)}{p_\theta(d)} \qquad \text{式 47}$$

由於$p_\theta(d)$為一個常數項，所以可以通過最大化$p_\theta(d|z)p_\theta(z)$的z值來找到最佳斷詞。如果假設的$p_\theta(z)$先驗機率是均勻分布，那麼可以透過下方式 48 獲得最佳斷詞z^*：

$$z^* = arg\max_z p_\theta(d|z) \qquad \text{式 48}$$

接著透過最大化對數概似函數L來優化模型的參數θ。如果邊緣化所有斷詞z，則可表示為式 49：

$$L(\theta) = \sum_d \ln p_\theta(d) = \sum_d \ln \sum_z p_\theta(d|z) \qquad \text{式 49}$$

在上述公式中，需要窮舉一個句子的所有可能斷詞的生成機率總和，但實際上是不可行的，因此需要使用動態規劃來計算。

在此可以利用 HMM 中所提到的前向-後向演算法來計算所有可能斷詞所生成的句子的總機率。首先，定義α為前向變數，t為字元序列的長度，k為預測的詞長度，簡短的字元序列$c_m \dots c_n$為$c_{m:n}$。然後，前向變數α_t^k表示字元序列$c_{1:t}$的最後k個字為單詞的機率。作者還使用$q_t = k$來表示長度為t的序列且其最後k個字為一個詞。因此有以下式 50 的推論：

$$\begin{aligned}\alpha_t^k &= p(c_{1:t}, q_t = k) = p(c_{1:t-k}, c_{t-k+1:t}, q_t = k) \\ &= \sum_{j=1}^{t-k} p(q_t = k, c_{t-k+1:t}, c_{1:t-k}, q_{t-k} = j) \\ &= \sum_{j=1}^{t-k} p(q_t = k|c_{t-k+1:t}) p(c_{t-k+1:t}|c_{1:t-k}) \alpha_{t-k}^j\end{aligned} \quad \text{式 50}$$

其中，$p(c_{t-k+1:t}|c_{1:t-k})$表示給定上文的片段字元序列生成機率，而$p(q_t = k|c_{t-k+1:t})$表示此片段為一個詞的機率。最後，可以通過正向變數來計算機率的總和：

$$\sum_z p_\theta(d|z) = \sum_{k=0}^{N} \alpha_N^k \quad \text{式 51}$$

其中，N是句子的長度，$\alpha_0^0 = 1$。

根據馬可夫性質，上述的機率與先前的狀態（即先前的斷詞和先前的上文）有關。所以，為了利用下文資訊進行斷詞，作者定義一個反向變數β_t^k，只需將原先句子反轉並再次使用式 51 即可獲得反向的機率總和。最後，可以對上述兩個機率進行平均，以充分利用上下文資訊，即可獲得目標函數$J(\theta)$，如式 52 所示：

$$\begin{aligned}J(\theta) = -L(\theta) &= -\sum_d \ln \sum_z p_\theta(d|z) \\ &= -\frac{1}{2}\sum_d \ln\left(\sum_{k=0}^{N_d} \alpha_{N_d}^k \sum_{k=0}^{N_d} \beta_{N_d}^k\right)\end{aligned} \quad \text{式 52}$$

前面幾個段落推導了非監督式學習框架如何去推斷最佳斷詞z，而後續則要講述如何應用上述框架於神經網路模型之中。$p(c_{t-k+1:t}|c_{1:t-k})$可以被神經語言模型所生成，可以定義可能的分割字元序列$c_{t-k+1:t}$為$s_{1:k}$，則可推導為下方式 53：

$$p(c_{t-k+1:t}|c_{1:t-k}) = \prod_{i=1}^{k} p(s_i|s_{1:i-1}, c_{1:t-k}) \quad \text{式 53}$$

在此作者在模型中使用了四個 LSTM，其中兩個分為前向和後向用於生成

上下文表示式，稱之為上下文 LSTM；剩下兩個也是分為前向和後向作為神經語言模型來生成上述機率，稱之為語言模型 LSTM 。以前向舉例來說，給定一個長度為N的字向量序列$c_1 \dots c_n$，上下文 LSTM f^C會生成上文的隱藏狀態$\vec{h}_1^C \dots \vec{h}_N^C$作為上文的表示式，而其隱藏狀態會用來初始化語言模型的 LSTM。在每個時間點t上下文 LSTM 的隱藏狀態$\vec{h}_1^C$透過下方式 54 更新：

$$\vec{h}_t^C = f^C(\vec{h}_{t-1}^C, c_{t-1}) \quad \text{式 54}$$

語言模型 LSTM f^L可以通過被訓練並經過 Softmax 函數去預測序列中下一個字元來學習可能的分割字元序列$s_{1:k}$的機率分布。語言模型 LSTM 的初始狀態$\vec{h}_0^L$可以從上下文 LSTM 在時間點$t-k+1$所生成的隱藏序列獲得，此隱藏序列可以看做是$s_{1:k}$的上文表示式，因此語言模型 LSTM 可以如下式 55、式 56、式 57 表示：

$$\vec{h}_0^L = \vec{h}_{t-k+1}^C \quad \text{式 55}$$

$$\vec{h}_{t'}^L = f^L(\vec{h}_{t'-1}^L, c_{t'-1}) \quad \text{式 56}$$

$$p(s_i|s_{1:i-1}, c_{1:t-k}) = \frac{\exp(w_j \vec{h}_i^L)}{\sum_{j'=1}^{V} \exp(w_{j'} \vec{h}_i^L)} \ (Softmax\ Func.) \quad \text{式 57}$$

最後，使用式 53 將這些機率組合起來，就可以得到給定上文的分割機率。相同的，後向的上下文 LSTM 和後向的語言模型 LSTM 也是如此。

$p(q_t = k|c_{t-k+1:t})$可以透過預測分割結束符號$< EOS >$的機率來建模，即$p(q_t = k|c_{t-k+1:t})$等價於$p(< EOS > |c_{t-k+1:t}, c_{1:t-k})$，$< EOS >$符號可以讓模型在預測是否成詞時與其他有意義的字競爭，降低錯誤成詞的機率。

解碼演算法部分，為了簡化公式，將$p(q_t = k|c_{t-k+1:t})$和$p(c_{t-k+1:t}|c_{1:t-k})$的乘積定義為給定w_i的上下文$ctx(w_i)$由字元序列$c_{t-k+1:t}$構成單詞w_i的機率$p_\theta(w_i|ctx(w_i))$。相同的，反向的版本也是如此定義為$p_\theta(w_i^*|ctx(w_i^*))$。通過將所有生成的單詞機率相乘來計算特定分割的機率，該特定分割將字元序列$c_1 \dots c_N$分割成單詞序列$w_1 \dots w_M$，並將反轉的字元序列$c_N \dots c_1$分割成$w_M^* \dots w_1^*$。

最後將$Z(d)$定義為句子d的所有可能斷詞的集合，接著計算下方式 58：

$$\underset{W,W^{*}\in Z(d)}{\operatorname{argmax}} \prod_{w_i\in W} p_\theta\left(w_i|ctx(w_i)\right) \prod_{w_i^{*}\in W^{*}} p_\theta\left(w_i^{*}|ctx(w_i^{*})\right) \quad \text{式 58}$$

但因為上式為 NP 問題，因此作者提出兩個動態規劃方法來解決它。

一、sgb-a：作者透過同一單詞的前向和後向機率進行平均，接著利用動態規劃來解碼出最佳斷詞。將$\hat{\alpha}_t^k$定義為字元序列$c_{1:t}$的最後k的字為單詞的最佳斷詞的機率，而δ_t^k被用來追溯解碼。

$$\bar{p}\left(w_j\right) = \sqrt{p_\theta(w_j|ctx(w_j))p_\theta(w_j^{*}|ctx(w_j^{*}))} \quad \text{式 59}$$

$$\hat{\alpha}_t^k = \max_{j=1}^{T} \bar{p}\left(w_j\right)\hat{\alpha}_{t-k}^j \quad \text{式 60}$$

$$\delta_t^k = \operatorname{argmax}_{j=1}^{T} \bar{p}\left(w_j\right)\delta_{t-k}^j \quad \text{式 61}$$

二、sgb-c：通過動態規劃分別使用兩側機率獲得最佳斷詞，並綜合所有單詞邊界以產生最終斷詞，從而獲得更細粒度的結果。

最後，作者在和之前大部分的的論文一樣都實驗於 SIGHAN 2005 的資料集上，並且根據 2011 年 Wang…等人提出的四種預處理設定[50]，作者使用了以下三種與其他模型進行比較：

1. 完全無監督式的訓練。
2. 所有標點符號都預先標示為分隔符號。
3. 所有標點符號和非中文字元都轉換成特殊符號表示。

因為中文裡對於單詞的概念並沒有很好的定義，因此不同的資料集根據不同的原則所構建。2007 年 Huang 和 Zhao…等人[51]估計了 SIGHAN 2005 中 4 個資料集之間的最低一致性比率0.848，而這個數值可以被視為非監督式模型的上限。作者所提出的方法結果已經非常接近這個上限。

第3章 結合非監督式與監督式學習之斷詞方法

3.1 動機以及目的

監督式中文斷詞在目前最廣泛使用的資料集 SIGHAN 2005 上已經獲得了巨大的成功，尤其在 2018 年 Google 提出了 BERT 的模型之後，模型在 SIGHAN 2005 各資料集上的 F1 分數（F1-Score）皆已經達到了 96 至 98 的高分數表現。因此，本論文繼續追求更高的 F1 分數已經有點吹毛求疵，在上一段的實驗中所提過的，在中文裡並沒有對單詞有很好的定義，在 SIGHAN 2005 的中文斷詞資料集中的 AS 和 CityU 兩個繁體中文資料集以及 MSR 和 PKU 兩個簡體中文資料集中標註方式並不一致，甚至在單一資料集中若人工標註的人員不同，也有可能標註出不同的正確答案，如：「台北市長」四個字，依據標註人員不同可以將它斷成「台北」、「市長」，也可以將它直接當成一個單詞。另外，現代的網路發達，每天不停地有新興的辭彙產生，有些學者提出可以透過在神經網路的架構上結合辭典的資訊讓模型可以在新興詞彙的斷詞上可以達到更好的效果，如 Qi Zhang 等人(2018)提出的 Hybrid LSTM 的方式[52]或 Junxin Liu 等人(2018)提出跨領域訓練及偽標記數據生成[53]來解決這樣的問題。監督式學習的方法無法隨時為模型更新資料集添加人工標註並重新訓練，以致造成模型在實際使用上的表現並不如在資料集上的測試結果來的好。

在以上這樣的情況下，再一味的追求更高的 F1 分數就顯得不切實際；相反地，非監督式學習的方法由於不需要正確標註讓模型學習，純粹利用最大化機率的方式計算最佳斷詞結果，就不存在於標註錯誤或不一致的問題。因此，最後決定採用非監督式學習之方式輔助監督式學習訓練，讓模型對於未出現過之詞語有更好的表現。

3.2 結合非監督式與監督式學習之斷詞框架

在本實驗中，監督式模型本論文使用BERT-CRF模型，因為監督式模型中，BERT 因為 Transformer 架構的緣故，詞向量可以根據上下文的不同而有不同的向量，能更有利於適應文章，CRF 則可以避免不合理標註結果的情況發生。而非監督式模型則採用雙向神經語言斷詞模型。首先，將資料集各自依照 BERT-CRF 模型以及雙向神經語言斷詞模型做前處理，並且分別輸入兩個模型之中，最後再將 BERT-CRF 模型輸出的一整句之 Loss 和雙向神經語言斷詞模型的整句之 Loss 進行依比例相加後再各自反向傳播回模型進行梯度更新，如 Figure 19 架構所示，而損失函數相加的算法如式 62 所示：

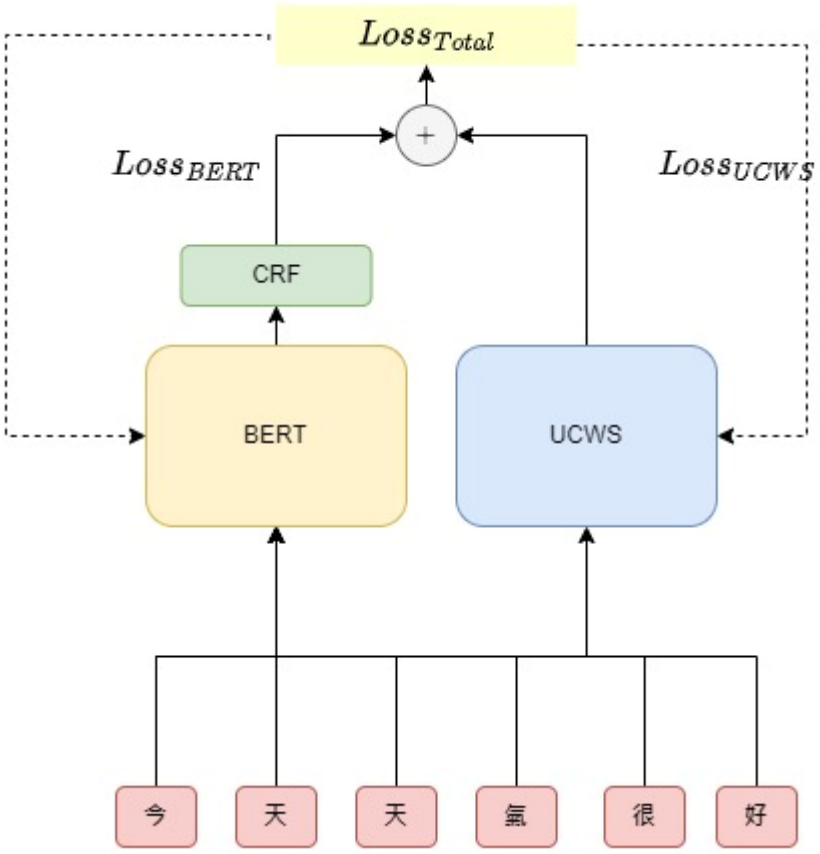

Figure 19　　模型架構圖

$$Loss_{total} = \alpha Loss_{BERT} + (1-\alpha) Loss_{UCWS} \quad \text{式 62}$$

其中，$Loss_{BERT}$和$Loss_{UCWS}$分別為 BERT-CRF 和雙向神經語言斷詞模型之損失函數，α為模型參數學習所得。

第4章 實驗設定

4.1 資料集

本論文使用資料集為 SIGHAN 2005 Chinese Word Segmentation Bakeoff [54]，其中又分為 4 個資料集分別為：

- AS（中研院）
- CityU（香港城市大學）
- MSR（微軟研究院）
- PKU（北京大學）

其中包含了訓練集、測試集、測試集標準答案、訓練集中抽取的詞表以及評分腳本，詞表主要用於評分時計算 OOV 比率（Out-Of-Vocabulary Rate）。

表格 5　　資料集統計

資料集	AS	CityU	MSR	PKU
語言	繁體中文	繁體中文	簡體中文	簡體中文
編碼	CP950	HKSCS	CP936	CP936
訓練集詞數	5,449,581	1,455,630	2,368,391	1,109,947
測試集詞數	122,610	40,936	106,873	104,372
詞表詞數	141,339	69,085	88,119	55,304

4.2 評估方式

首先，介紹機器學習模型的幾個模型評估指標：精確率（Precision）、召回率（Recall）和 F 分數（F1-Score 或稱 F1-Measure）[55]，其可用一個二分類的

混淆矩陣來說明如下表格 6 所示：

表格 6　　混淆矩陣

真實值/預測值	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

1. True Positive（TP）：真實為真且預測也為真的情況，模型預測正確。
2. False Negative（FN）：真實為真但預測為假的情況，模型預測錯誤。
3. False Positive（FP）：真實為假但預測為真的情況，模型預測錯誤。
4. True Negative（TN）：真實為假且預測也為假的情況，模型預測正確。

根據混淆矩陣，可以定義出精確率和召回率的計算方式：

- 精確率：預測為真的樣本中真實也為真的樣本，可以想像成模型預測為真的樣本中預測正確的有多少個。

$$Precision = \frac{TP}{TP + FP} \quad \text{式 63}$$

- 召回率：真實為真的樣本中預測也為真的樣本，可以想像成真實為 z 的樣本中，能被模型找回多少個。

$$Recall = \frac{TP}{TP + FN} \quad \text{式 64}$$

最後，中文斷詞模型的好壞主要透過 F1 分數來評估模型效能，F1 分數是一個綜合衡量精確率以及召回率的評估指標。F1 分數之數學式如下式 65 所示：

$$F - Score = (1 + \beta^2)\frac{Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \quad \text{式 65}$$

當$\beta = 1$時，即 F1 分數，也就是召回率（Recall）和精確率（Precision）的調和平均值。

4.3 實驗設定

4.3.1 預訓練字向量

在本論文中，Stacked－CNN CRF、BiLSTM CRF 以及 SLM 模型中皆須使用預訓練字向量，Stacked－CNN CRF 模型使用 Word2Vec，而 BiLSTM CRF 使用 Wang2Vec。繁體中文訓練於 Chinese Wikipedia Gigaword Corpus，簡體中文訓練於搜狗語料庫。

表格 7　　字向量訓練參數設定表

參數設置	Word2Vec	Wang2Vec
字向量模型	Skip-gram	Skip-gram
學習率	0.025	0.05
Epoch 次數	5	5
N-gram 大小	1-4	1-2
Windows 大小	1-5	1-2
詞頻捨棄大小	10	-

4.3.2 基礎實驗模型設定

在本論文實驗中使用模型設定如下：

1. Stacked－CNN CRF：使用上一節所示預訓練字向量維度為 200，詞向量維度為 50，總共包含 5 層卷積層，每層通道大小為 200 維，卷積核寬度為 3，在嵌入層及每層卷積層皆使用丟失率（Dropout）設為 0.2，在每層卷積層使用 Batch Normalization，批次大小設為 100，優化器使用 Adam，訓練次數不超過 100 次，最後模型取自於發展集最優結果。

2. BiLSTM CRF/Softmax：使用上一節所示預訓練字向量維度為 256，Bigram 字向量維度為 64，總共有兩層維度為 256 的雙向 LSTM，LSTM 的丟失率設為 0.2，學習率為[0.04, 0.035, 0.03]，每 32000 步下降 0.005。
3. BERT CRF/Softmax：使用 Google 釋出預訓練模型 Chinese-bert-base，輸入長度限制 512，共使用 12 層維度 768 的 Transformers 編碼器，12 個頭的多頭注意力機制。
4. 分段語言模型（SLM）：使用上一節所示預訓練字向量維度為 256，編碼器與解碼器的 LSTM 維度為 256，最大詞長度為[2, 3, 4] ，資料前處理會將標點符號、英文字母以及阿拉伯數字轉換成特殊符號。
5. 雙向語言斷詞模型（UCWS）：使用上一節所示預訓練字向量維度為 300 或是不使用預訓練字向量，編碼器與解碼器 LSTM 維度各為 300，最大詞長度為[3, 4, 5]，資料前處理會將標點符號、英文字母以及阿拉伯數字轉換成特殊符號，訓練次數不高於 40 次。

第5章 實驗結果與討論

5.1 基礎系統

本章節呈現所有基礎模型於各資料集之 F1 分數，如下方表格 7 所示，表格上方區域為監督式學習之基礎模型，下方為非監督式學習基礎模型，在非監督式模型中數字代表限制單詞最大長度K，而 UCWS 模型中，又依照解碼方式不同分為 sgb-a 和 sgb-c。

表格 8　　基礎模型於 4 個資料集之 F1 分數

模型/資料集	AS	CityU	MSR	PKU
Stacked – CNN CRF	96.6	94.5	98.1	**97.4**
BiLSTM CRF	95.6	95.5	96.7	94.5
BERT CRF	96.7	97.8	98.4	96.6
BERT Softmax	**96.9**	**97.9**	**98.5**	96.7
SLM-2	79.2	78.7	77.6	80.0
SLM-3	79.9	**80.5**	80.5	**80.7**
SLM-4	79.7	80.5	79.0	79.8
UCWS sgb-a-3	81.7	79.9	82.3	79.1
UCWS sgb-a-4	81.6	79.7	81.0	77.8
UCWS sgb-a-5	81.4	77.7	79.1	75.1
UCWS sgb-c-3	79.1	79.6	**82.5**	80.5
UCWS sgb-c-4	82.3	80.1	81.6	78.0
UCWS sgb-c-5	**82.4**	80	81.6	78.8

可以看到 4 個基礎監督式模型的分數並沒有很大的差異，若要純粹以 F1 分數來評斷每個模型的好壞是有困難的，就如本論文前一章節動機所說，現在的斷詞模型皆能達到非常好的表現，但在實際應用時通常效果沒有這麼的好。在本論文提出的模型上，監督式部分本論文選用 BERT CRF，因為 BERT 的模型式利用 Transformer 的架構，並無如 RNN、GRU、LSTM…等遞迴神經網路會因時

間拉長導致訊息遺失，BERT 的字向量會根據上下文的不同擁有不同的表示法且已經預訓練在非常大量的文本上，在實際時用會相較 Word2Vec 預訓練的字向量更容易去適應各式文章；Softmax 與 CRF 從 F1 分數上並無何者表現特別突出，但因為 CRF 的當前隱藏狀態會考慮到前一個狀態，表示可以避免如：BM 後面卻接著 S 的這種標註形式產生，故選用 CRF。

而非監督式學習的基礎模型部分，可以看到無論最大單詞長度K為何，雙向語言斷詞模型 UCWS 大致優於或持平只利用單向資訊的 SLM 模型，但並無法找到何種K最適合用於斷詞，因此非監督式的模型本論文選用單一資料集中 F1 分數最高之K值模型，AS 使用 sgb-c-5，CityU 使用 sgb-c-4，MSR 使用 sgb-c-3，PKU 使用 sgb-c-3。

5.2 結合非監督與監督式學習之斷詞模型實驗

在本章節將呈現本論文所提出的方法之實驗結果，在此稱之為 ***U+BERT*** 模型，如下表格 9 所示。

表格 9　　U+BERT 實驗結果

模型	AS	CityU	MSR	PKU
UCWS	82.4	80.1	82.5	80.5
BERT CRF	96.7	97.8	98.4	96.6
U+BERT	***96.1***	***96.9***	***97.6***	***95.9***

從實驗中可以看到，提出的模型在 F1 分數上並沒有超越原有的 BERT CRF 模型，推測原因可能出自於非監督式的斷詞並未達到有效的輔佐 BERT CRF 模型。

在本論文第三章有提到，兩個模型的$Loss_{BERT}$和$Loss_{UCWS}$之前有經過一個可訓練變數α來控制每個$Loss$的比例為多少。在此實驗中，α最後的值為0.8－

0.9區間，表示 UCWS 所提供的輔助並不多，甚至導致 BERT CRF 的模型分數稍稍下降。因此從實驗結果中推測主要原因出自於非監督式模型並不夠強大， 不足以對 BERT 模型造成夠多的正面影響。

表格 10　　U+BERT OOV Recall Rate

模型	AS	CityU	MSR	PKU
BERT CRF	80.5	88.2	87.5	87.2
U+BERT	***80.1***	***88.1***	***86.4***	***86.6***

表格 10 為實驗於四個資料集上的 OOV Recall Rate 結果，可以發現本論文所提出的方法於 AS、MSR、PKU 資料集上的 OOV Recall Rate 皆略低於基準模型 BERT CRF 模型，而於 CityU 的資料集卻能約略與其持平，因此本論文所提出之方法可能對於 OOV 的處理上有些許作用，但並未達到本研究之預期。

5.3 CNN 與 BiLSTM 模型之探討

在本章節將討論傳統預訓練字向量 Stacked－CNN 與 BiLSTM 模型中，若未對輸入資料的標點符號、阿拉伯數字、中文數字及英文進行前處理（如：非監督式方法裡替換成特殊符號），最後結果通常會導致預測錯誤，如下表格 11 所示：

表格 11　　標註錯誤範例

原始句子	預測標註	正確標註
最低三五・〇五元，	最/低/三/五/・/〇/五/元/，/	最/低/三五・〇五/元/，/
英式英語用 center，	英式/英語/用/ce/nt/er/，/	英式/英語/用/center/，/

因此，本論文對這部分做了一個小實驗於 AS 資料集上，統計這些類型的錯誤在資料集中是否占了很大比例，如下表格 12、表格 13、表格 14、表格 15 所示：

表格 12　　基準模型數據

	模型	Precision	Recall	F-score
基準模型	Stacked – CNN CRF	95.4	97.9	96.7
	BiLSTM CRF	95.1	96.0	95.6
	BERT-Softmax	96.6	97.0	96.8
	BERT CRF	96.4	97.0	96.7

表格 13　　去除 AS 資料集中文字元後之數據

	模型	Precision	Recall	F-score
去除中文字元	Stacked – CNN CRF	97.2	99.0	98.1
	BiLSTM CRF	97.3	98.4	97.9
	BERT-Softmax	99.2	98.8	99.0
	BERT CRF	99.1	98.8	99.0

表格 13 中可以看到 BERT 的模型在資料集去除了所有中文字之後，明顯在處理英文及數字上有很好的表現，而 Stacked – CNN 與 BiLSTM 模型雖然不算特別差，但是和 BERT 模型在英文及數字上有了約 2 個百分點的差距，整個資料集的字元數為 226537 個，而去除中文字元後剩下的字元數為 54342 個，其中大多為標點符號，英文和數字的占比並不高，所以對整體 F1 分數影響有限。

表格 14　　去除 AS 資料集英文、數字及標點符號之數據

	模型	Precision	Recall	F-score
去除英文、數字及標點符號	Stacked – CNN CRF	95.4	97.8	96.6
	BiLSTM CRF	95	95.8	95.4
	BERT-Softmax	96.5	97	96.8
	BERT CRF	96.3	96.9	96.6

表格 14 中讓所有模型只專注於在中文字元上，所有模型的表現與基礎模型相差不遠，更凸顯出上一段的結論。

最後，本實驗試著使用了後處理的方式再對上述問題進行處理，將相鄰英文字元及數字（包含阿拉伯數字以及中文數字）進行合併並與其相鄰中文字元分開，實驗數據如下表格 15 所示：

表格 15　　後處理後之數據

	模型	Precision	Recall	F-score
後處理	Stacked – CNN CRF	95.5	97.8	96.7
	BiLSTM CRF	95.0	95.6	95.3
	BERT-Softmax	96.0	96.8	96.4
	BERT CRF	96.2	96.9	96.6

從此實驗中最後的結果來看，後處理後 F1 分數不升反降，因為資料集中有許多如：「1999 年」經過後處理後雖然能讓 1999 合併成了一個詞，但必定會與後方之「年」分隔開，或者如「戴维·米拉克」則會因為中間的標點符號，讓模型原本判斷正確的人名又因此被斷開。因此，後處理的結果不盡理想，因為後處理需採用 Rule-base 的方式，但總有規則中的漏洞無法涵蓋。

5.4 雙向語言斷詞模型之探討

本章節將探討非監督式學習的雙向語言斷詞模型在訓練上可能導致效能低落之原因以及架構上修改成果。在該實驗中，本實驗將編碼器及解碼器各替換成 LSTM、GRU 以及 RNN 的所有配對組合，並訓練於 SIGHAN 2005 的四個資料集上；除此之外，也將會呈現模型架構改為各式遞迴神經網路結構和 Transformers 的成果以及探討其失敗原因。在此將 LSTM 簡稱為 L，GRU 簡稱為 G，而 RNN 簡稱為 R。

首先，雙向語言斷詞模型訓練時容易因為輔助詞沾黏的問題，導致模型效能低落，例如：「了」、「的」這些詞語沾黏到了前面的詞語中。訓練過程中並不能完全保證模型最後能收斂到輔助詞被切開的情況，所以不同的參數初始化可能有些效果很好，而有些直到迭代結束依然難以切開沾黏詞。 因此，若是在第一個迭代結束時確認發現驗證集存在有輔助詞粘連的問題，模型就必須重新初始化參數並重新訓練。又或者使用更暴力一點的方法，在替換標點符號及數字…等特殊標誌時，將容易產生粘連問題的輔助詞一併加入，將其視為一種必要的斷點，最後在輸出結果時再加回文本中。

表格 16　　遞迴結構測試於 AS 資料集

AS	L-L	L-G	L-R	G-L	G-G	G-R	R-L	R-G	R-R
Sgb-a-3	81.7	**81.8**	81.2	**81.2**	31.6	77.4	80.7	**80.3**	72.0
Sgb-a-4	79.1	73.4	81.2	80.3	80.4	81.6	79.9	79.2	69.8
Sgb-a-5	81.6	81.5	81.6	80.3	**81.4**	78.6	79.7	77.9	61.2
Sgb-c-3	82.3	79.2	**82.2**	77.2	39.6	79.9	82.1	40.4	**75.4**
Sgb-c-4	81.4	81.7	80.8	80.9	40.9	**81.7**	**82.3**	59.0	73.1
Sgb-c-5	**82.4**	78.5	81.9	44.2	41.1	79.7	82.2	69.5	72.9

表格 17　　遞迴結構測試於 CityU 資料集

CityU	L-L	L-G	L-R	G-L	G-G	G-R	R-L	R-G	R-R
Sgb-a-3	79.9	**80.7**	80.8	79.5	**79.5**	80.0	**79.2**	**79.1**	79.4
Sgb-a-4	79.6	79.7	**81.1**	**80.0**	78.5	80.0	79.0	78.5	**79.9**
Sgb-a-5	79.7	78.7	80.8	77.8	77.8	78.3	76.3	76.4	78.5
Sgb-c-3	**80.1**	79.4	80.9	78.8	78.3	80.0	79.0	78.4	79.1
Sgb-c-4	77.7	78.0	77.1	75.8	77.1	78.3	74.5	74.3	75.8
Sgb-c-5	80.0	79.2	80.3	78.8	78.8	80.0	78.7	76.4	78.1

表格 18　　遞迴結構測試於 MSR 資料集

MSR	L-L	L-G	L-R	G-L	G-G	G-R	R-L	R-G	R-R
Sgb-a-3	82.3	**82.8**	81.2	81.1	81.5	82.5	78.7	78.7	79.0
Sgb-a-4	**82.5**	82.5	**83.2**	**82.1**	76.7	79.4	42.3	42.3	79.0
Sgb-a-5	81.0	78.9	79.9	78.7	65.9	77.0	36.7	38.9	73.5
Sgb-c-3	**82.5**	81.6	81.6	80.8	**81.8**	**83.3**	**82.1**	**82.1**	**79.4**
Sgb-c-4	79.1	79.9	81.6	65.5	80.5	81.0	76.8	76.8	75.8
Sgb-c-5	82.2	81.6	80.8	80.0	77.7	78.1	71.5	51.9	73.9

表格 19　　遞迴結構測試於 PKU 資料集

PKU	L-L	L-G	L-R	G-L	G-G	G-R	R-L	R-G	R-R
Sgb-a-3	79.1	78.3	79.9	78.4	**79.0**	78.7	78.5	77.9	74.3
Sgb-a-4	**80.5**	**80.6**	**80.2**	**79.4**	75.6	77.5	75.6	72.4	75.5
Sgb-a-5	77.8	75.1	75.8	76.1	71.8	75.3	72.7	40.9	75.3
Sgb-c-3	78.0	78.6	78.7	**79.4**	78.4	**79.1**	**78.8**	**79.2**	**78.9**
Sgb-c-4	75.1	63.5	66	77.9	79.0	78.9	76.2	74.7	78.2
Sgb-c-5	78.8	77.6	77.7	77.4	78.6	76.8	75.6	72.7	75.7

在此章節的實驗中，首先可以看到無論在哪一個資料集或哪一個架構上，紅字部分為前一段落所提到的輔助詞沾黏問題非常嚴重，模型從初始至結束收斂的不是很好，因此導致模型訓練的穩定性不甚好，效能起伏非常大。

在最大詞長度K的比較下，在 PKU 資料集中可以發現，不論在何種架構下，K值為 4 之模型普遍分數較高，而其他資料集中，雖然無法明顯看出何種K值最佳，但可以得知因為中文的詞較少有 5 個字的情況，因此$K = 5$的狀況下模型雖然也可以斷出$1-4$字的詞，但因為增加了可能有五個字成詞的機率，而稍微拉低了一點分數，以致$K = 5$的模型通常不是最佳。另外，因此本實驗中有統計出每個資料集中$1-5$字成詞的數量，如表格 20。大於或等於五個字以上的通常為地名、人名、英文及數字類，如：「中華人民共和國」、「2000 年」…等等。

表格 20　　資料集字詞統計表

字詞數\資料集	AS	CityU	MSR	PKU
一字詞	41,000	17,341	43,257	46,891
二字詞	45,241	17,365	47,659	48,627
三字詞	11,578	2,830	5,566	5,972
四字詞	8,201	1,339	4,347	2,111
五字詞	1,480	385	1,136	597
五字以上	691	203	1,316	174

表格 21　　BERT 架構之 UCWS

資料集/模型	Sgb-a-3	Sgb-a-4	Sgb-a-5	Sgb-c-3	Sgb-c-4	Sgb-c-5
CityU	55.5	49.4	42.7	41.8	50.2	43.5

現今的論文中，大部分 Transformer 的架構都能比遞迴神經網路效果更好，因此若將 UCWS 上下文編碼器中的 LSTM 替換成 Transformer 的架構，似理論上能獲得更豐富的上下文資訊。表格 21 為將 UCWS 中上下文編碼器的 LSTM 直接

替換成 BERT 架構之實驗結果，可以明顯看出模型幾乎無法訓練，推測原因為在原作者的訓練架構上，雖說使用雙向 LSTM 去獲取上下文資訊，但實際上訓練時作者使用的不是大家所常見的 BiLSTM 架構直接獲取一個句子的正反向資訊，而是透過兩個前向 LSTM 分別對句子進行反轉獲取上下文資訊並分別計算其機率後再合併。本研究探究其原因出自於 LSTM 之隱藏狀態會拿去當解碼器之初始狀態，替換成 BERT 之後解碼器之初始狀態包含了前後向之資訊，導致語言模型還擁有了未來資訊，因此將模型替換成 BERT 架構效能低落可能與此有關。這一部分的問題是值得去探討的地方，若能解決或能增進非監督式斷詞之效能。

5.5 常用套件比較

- 結巴斷詞：中文斷詞最常見的套件，使用規則斷詞，主要是透過詞典，在對句子進行斷詞的時候，將句子的每個字與詞典中的詞進行匹配，找到則斷詞，否則無法斷詞。對於未登錄詞的部分，則會採用 HMM 的方式來推斷是否成詞。
- Ckiptagger： 目前效能最好之斷詞套件，採用 BERT、ALBERT、GPT-2…等 Transformer 模型架構。
- 交大語音實驗室：使用 CRF 模型及中研院 Sinica Corpus 語料庫經大量錯誤更正後訓練發展而成。
- Monpa[56]：採用 BERT-CRF 模型架構，目前最新架構改為 ALBERT-CRF。

表格 22　　常用套件比較表

斷詞系統	AS
繁中版—Jieba	88.6
Ckiptagger	97.7
交大語音實驗室	95.6
Monpa	93.7
U+BERT	***96.1***

5.6 基礎模型優缺點

- Stacked－CNN CRF：模型訓練速度快，F1 分數能達到不錯的結果，但需要訓練教師模型以及學生模型，且需使用額外未標註資料，訓練步驟較繁雜。對於英文字母、數字、標點符號的斷詞結果較差。
- BiLSTM CRF：模型訓練速度慢，但不須額外資訊且訓練步驟簡單，而能達到不錯的 F1 分數，對於英文字母、數字、標點符號的斷詞結果較差。
- BERT CRF/Softmax：使用預訓練的模型微調下游任務，因此儘管餐數量龐大但訓練速度介於 BiLSTM CRF 和 Stacked－CNN CRF 之間，且 F1 分數為所有模型中最高。
- SLM：因為需要計算句子中每個字的成詞機率，因此訓練速度較監督是學習方法慢，對於英文字母、數字、標點符號有使用前處理，在非監督式方法能達到次高的 F1 分數。
- UCWS：訓練速度略慢於 SLM，但因其使用雙向語言模型，對於英文字母、數字、標點符號有使用前處理在非監督式方法中達到最高的 F1 分數。
- U+BERT：使用監督式與非監督式方法結合，訓練速度相同於 UCWS，F1 分數略差於 BERT CRF 模型。

第6章 結論與未來展望

在中文斷詞的領域中，目前現有的模型中大部分都能取得非常好的效能，因此，本研究認為評估於特定資料集的 F 分數高低已經不適用於現今的斷詞模型，或許需要有更鑑別性的資料集的提出，讓模型能更有效的應用於現今的網路文章、新興詞彙…等。

此論文提出了一個結合監督式學習與非監督式學習的方法，雖然實驗結果只取得和基礎模型相近的分數，但從本論文的實驗結果顯示此種學習方式可能會是有效的，尤其是對於非監督式學習的模型，如果能改善非監督式方法的效能，讓監督式學習的模型能在於某些不確定詞彙更依賴於非監督式方法，如此就能更有效處理新詞的問題。

雖然在監督式的方法中似乎模型都到達了難以突破的瓶頸，但非監督式方法中尚有改善的空間，如本論文中所提到的利用 Transformer 架構改善原有模型或許能有不錯的效果。

參考文獻

[1] F. Stahlberg, "Neural machine translation: A review," *Journal of Artificial Intelligence Research,* vol. 69, pp. 343-418, 2020.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR 2015,* 2015.

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Learning to compose neural networks for question answering," *arXiv preprint arXiv:1601.01705,* 2016.

[4] A. M. N. Allam and M. H. Haggag, "The question answering systems: A survey," *International Journal of Research and Reviews in Information Sciences (IJRRIS),* vol. 2, no. 3, 2012.

[5] A. Nenkova and K. McKeown, "Automatic summarization," *Foundations and Trends® in Information Retrieval,* vol. 5, no. 2–3, pp. 103-233, 2011.

[6] Y. Liu, "Fine-tune BERT for extractive summarization," *arXiv preprint arXiv:1903.10318,* 2019.

[7] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing,* vol. 22, no. 10, pp. 1533-1545, 2014.

[8] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*, 2013: Ieee, pp. 6645-6649.

[9] Y. Lin, H. Ji, F. Huang, and L. Wu, "A joint neural model for information extraction with global features," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7999-8009.

[10] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Part-of-speech tagging with bidirectional long short-term memory recurrent neural network," *arXiv preprint arXiv:1510.06168,* 2015.

[11] X. Zheng, H. Chen, and T. Xu, "Deep learning for Chinese word segmentation and POS tagging," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 647-657.

[12] Y. Tian, Y. Song, F. Xia, T. Zhang, and Y. Wang, "Improving Chinese word segmentation with wordhood memory networks," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 8274-8285.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature,* vol. 323, no. 6088, pp. 533-536, 1986.

[14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science,* vol. 313, no. 5786, pp. 504-507, 2006.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60,

no. 6, pp. 84-90, 2017.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," ***NIPS 2014 Deep Learning and Representation Learning Workshop,*** 2014.

[19] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks."

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing,* vol. 45, no. 11, pp. 2673-2681, 1997.

[21] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ICLR 2013,* 2013.

[23] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Advances in neural information processing systems,* vol. 13, 2000.

[24] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.

[25] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational*

linguistics, vol. 5, pp. 135-146, 2017.

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems,* vol. 26, 2013.

[27] M. N. Matthew E. Peters, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer, "Deep contextualized word representations," presented at the NAACL 2018, 2018.

[28] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL 2019,* 2019.

[30] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942,* 2019.

[31] M. Lewis *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics,* 2020.

[32] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *NeurIPS 2019,* 2019.

[33] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: a compact task-agnostic bert for resource-limited devices," *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, 2020.

[34] Y. Liu *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692,* 2019.

[35] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems,* vol. 32, 2019.

[36] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for chinese bert," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 29, pp. 3504-3514, 2021.

[37] N. Xue, "Chinese word segmentation as character tagging," in *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, 2003, pp. 29-48.

[38] C. Huang and H. Zhao, "Which is essential for Chinese word segmentation: Character versus word," in *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, 2006, pp. 1-12.

[39] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning hidden Markov model structure for information extraction," in *AAAI-99 workshop on machine learning for information extraction*, 1999, pp. 37-42.

[40] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* vol. 77, no. 2, pp. 257-286, 1989.

[41] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy Markov models for information extraction and segmentation," in *Icml*, 2000, vol. 17, no. 2000, pp. 591-598.

[42] F. Peng, F. Feng, and A. McCallum, "Chinese segmentation and new word detection using conditional random fields," in *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, 2004, pp. 562-568.

[43] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[44] J. Ma, K. Ganchev, and D. Weiss, "State-of-the-art Chinese word segmentation with Bi-LSTMs," *Association for Computational Linguistics,* 2018.

[45] D. Weiss, C. Alberti, M. Collins, and S. Petrov, "Structured training for neural network transition-based parsing," *Association for Computational Linguistics,* 2015.

[46] C. Wang and B. Xu, "Convolutional neural network with word embeddings for Chinese word segmentation," *Asian Federation of Natural Language Processing,* 2017.

[47] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, 2017: PMLR, pp. 933-941.

[48] Z. Sun and Z.-H. Deng, "Unsupervised neural word segmentation for chinese via segmental language modeling," *Association for Computational Linguistics,*

2018.

[49] L. Wang and X. Zheng, "Unsupervised word segmentation with bi-directional neural language model," *ACM Transactions on Asian and Low-Resource Language Information Processing,* vol. 22, no. 1, pp. 1-16, 2022.

[50] H. Wang, J. Zhu, S. Tang, and X. Fan, "A new unsupervised approach to word segmentation," *Computational Linguistics,* vol. 37, no. 3, pp. 421-454, 2011.

[51] C. Huang and H. Zhao, "Chinese word segmentation: A decade review," *Journal of Chinese Information Processing,* vol. 21, no. 3, pp. 8-20, 2007.

[52] Q. Zhang, X. Liu, and J. Fu, "Neural networks incorporating dictionaries for Chinese word segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.

[53] J. Liu, F. Wu, C. Wu, Y. Huang, and X. Xie, "Neural chinese word segmentation with dictionary knowledge," in *CCF International Conference on Natural Language Processing and Chinese Computing*, 2018: Springer, pp. 80-91.

[54] T. Emerson, "The second international Chinese word segmentation bakeoff," in *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*, 2005.

[55] Y. Sasaki, "The truth of the F-measure," *Teach tutor mater,* vol. 1, no. 5, pp. 1-5, 2007.

[56] W.-C. Yeh, Y.-L. Hsieh, Y.-C. Chang, and W.-L. Hsu, "MONPA: 中文命名實體及斷詞與詞性同步標註系統 (MONPA: A Multitask Chinese

Segmentation, Named-entity and Part-of-speech Annotator)," in *Proceedings of the 31st Conference on Computational Linguistics and Speech Processing (ROCLING 2019)*, 2019, pp. 241-245.