正體中文斷詞系統應用於大型語料庫之多方評估研究
Multifaceted Assessments of Traditional Chinese Word Segmentation Tool on Large Corpora

Wen-Chao Yeh
Institute of Information Systems and Applications
National Tsing Hua University
Taiwan
wyeh@m109.nthu.edu.tw

Yu-Lun Hsieh
Graduate Institute of Data Science
Taipei Medical University
Taiwan
morpheus.h@gmail.com

[1] Yung-Chun Chang
Graduate Institute of Data Science
Taipei Medical University
Taiwan
changyc@tmu.edu.tw

Wen-Lian Hsu
Department of Computer Science and Information Engineering
Asia University
Taiwan
Pervasive AI Research Labs
Ministry of Science and Technology
Taiwan
hsu@iis.sinica.edu.tw

摘要

本研究之目的在於運用多種數值指標及實驗資料來評估 CKIP、Jieba、MONPA 等三種廣泛應用於臺灣自然語言處理產學界的正體中文斷詞器。我們特別針對運算效能、資源需求等等面向，檢驗其應用於大型語言文字資料集時，處理斷詞、詞性標註及命名實體辨識等工作之成效。實驗結果顯示，MONPA 利用圖形運算加速器（GPU）進行批次處理斷詞時，可以大幅度縮減巨量中文資料的運算時間，且其斷詞、詞性標註、命名實體辨識等多功能標籤均達到令人滿意的品質，且其產出之標註結果可有效輔助提升中文自然語言處理的後續相關任務成效。

Abstract

This study aims to evaluate three most popular word segmentation tool for a large Traditional Chinese corpus in terms of their efficiency, resource consumption, and cost. Specifically, we compare the performances of Jieba, CKIP, and MONPA on word segmentation, part-of-speech tagging and named entity recognition through extensive experiments. Experimental results show that MONPA using GPU for batch segmentation can greatly reduce the processing time of massive datasets. In addition, its features such as word segmentation, part-of-speech tagging, and named entity recognition are beneficial to downstream applications.

關鍵字：自然語言處理，中文斷詞，詞性標註，命名實體辨識
Keywords: NLP, Chinese Word Segmentation, POS, NER

1 緒論

近幾年來人工智慧應用發展可說是突飛猛進，但據我們觀察，可以處理正體中文的人工智慧模型仍存在進步空間，主要肇因於中文自然語言處理（NLP）的基礎設施仍未到

[1] Corresponding author

位。其中，特別是斷詞（Word Segmentation）這個自然語言處理流程中一個重要步驟，因有別於英文書寫上可用空白（white space）為線索來找到詞彙的邊界，中文書寫系統中的空白並不帶有任何詞語邊界的意義。正因為中文可以將單字或多字視為一個詞彙，要使用計算機來分析、擷取中文的資訊，就需要先以特殊工具完成斷詞處理。綜觀現今國內外產學界在中文自然語言處理，我們歸納出最常用來處理正體中文斷詞的工具為 MONPA[2]、CKIP[3]、Jieba[4] 等三種。

一般認為 Jieba 斷詞系統速度較快，但正確率較低；CKIP 最新版本增加開發了 python 套件，保持其長久以來優良的成效，且更方便使用。 MONPA 對正體中文的支援度與 CKIP 處於伯仲之間。然而，至今尚未有嚴謹的學術研究針對這三種工具作完整的評測實驗。故本文將以上述三種斷詞工具對正體中文的斷詞、詞性標註、命名實體辨識等功能做多樣化的性能分析研究。更詳細來說，我們將實驗並紀錄三種工具的套件載入及斷詞運行時間，再分別以 SIGHAN 歷年來多筆 Share Task 的公開資料集，與專業人工標註的新聞語料等資料進行正確率驗證。最後，從網際網路以爬蟲技術搜集大量資料集供做文本分類任務使用，以驗證不同斷詞工具的斷詞結果是否影響機器學習的分類表現。

綜合實驗結果顯示，MONPA 利用 GPU 施行批次斷詞處理，可以大幅度縮減巨量中文資料的斷詞時間，且其斷詞、詞性標註、命名實體辨識等成果亦具有相當可靠的品質，有益於後續應用機器學習作中文自然語言處理的相關任務。

2 研究方法

Jieba 是基於簡體中文語料，透過 HMM 模型 (Baum et al., 1970) 所訓練出來的工具。就原始版本而言，對正體中文的支援度不佳，但可透過手動載入正體字詞字典檔來改善斷詞效果。CKIP 為歷史悠久的斷詞工具，經中研院 CKIP Lab 以較新穎的 BiLSTM 架構訓練模型 (Li et al., 2020)，並以 Python 套件釋出。MONPA (Hsieh et al., 2017) 最初為基於遞歸神經網路 (Recurrent Neural Network, RNN) 所建立的模型，並包含雙向 (bidirectional) 結構以便學習更廣泛的語境知識，同時也引入注意力 (attention) 機制，達到更佳的斷詞、標註等效果。

除了所採用的模型及理論基礎不同以外，這三種斷詞工具在工程層面亦有所差異，因此運行的環境也不盡相同。除 Jieba 採用自行開發的程式框架，CKIP 利用了 Tensorflow 這個深度學習工具庫為基礎架構，MONPA 則是採用 Pytorch 架構開發。為了盡可能的降低環境變因，所以，本研究的實驗環境將基於同一硬體設備，以 conda 建構 python 運行環境。我們的硬體設置如下：

- CPU: 4 * AMD EPYC 7252 8-Core Processor
- GPU: 7 * NVIDIA GeForce RTX 3090 (24GB memory)
- Memory: 8 * 32 GB (DDR4 3200 MT/s)
- OS: Ubuntu 20.4 LTS

2.1 斷詞工具版本

- Jieba：安裝 0.42.1 版本[5]，並另外下載約 4 MB 大小的正體中文詞典。實驗時將分別測試：(1) 預設版本，後稱 Jieba； (2) 匯入正體中文字典檔版本，後稱 JiebaD。
- CKIP：安裝 0.2.1 版本[6]，並另外下載約 1.8 GB 大小的模型檔，運行於 Tensorflow 2.6.0 架構。後稱 CKIP。
- MONPA：安裝 0.3.3 版本[7]（內含 8.9 MB 大小的模型檔），運行於 Pytorch 1.11.0 架構。實驗時將分別測試：(1) MONPA 預設的單句斷詞方法，後稱 MONPA；(2) 應用 GPU 效能的批次斷詞方法，後稱 MONPA Batch。

2.2 評測項目

本研究將實驗上述三種斷詞工具對正體中文的斷詞、詞性標註、命名實體辨識等功能的成果及效率。首先，我們紀錄三種工具的套件載入及斷詞運行時間，再分別以 SIGHAN (AFNLP, 2003; AFNLP, 2005; Ng & Kwong, 2006; AFNLP, 2008) 歷年來多筆 Share Task 的公開資料集，及經過專業人工標註的新聞語料等資料集驗證。另外，我們也從網際網路

[2] https://github.com/monpa-team/monpa/
[3] https://github.com/ckiplab/ckiptagger
[4] https://github.com/fxsjy/jieba
[5] https://pypi.org/project/jieba/
[6] https://pypi.org/project/ckiptagger/
[7] https://pypi.org/project/monpa/

以爬蟲搜集三種文本分類任務的資料集，以檢驗不同斷詞工具的斷詞結果，是否會影響機器學習的分類表現。

2.2.1 斷詞執行效率

此實驗分三階段測試斷詞工具的執行效率，依序為：載入套件時間、一千句內的小資料集斷詞時間、5,000 句至 40,000 句的大資料集斷詞時間。運行時間以 python 基本套件 time 執行紀錄，每筆測資皆運行 10 次取平均值。

- 載入套件時間：三種斷詞工具皆是 python 套件，因此本次實驗將先紀錄斷詞工具的套件載入需要費時多久。
- 小資料集斷詞時間：每一句皆是由 200 個字元長度組成，實驗從一句到 990 句的斷詞執行時間各需多久。
- 大資料集斷詞時間：每一句皆是由 200 個字元長度組成，分別測試對 5,000 句、10,000 句、15,000 句、20,000 句、25,000 句、30,000 句、35,000 句、 40,000 句的斷詞執行時間各需多久。

2.2.2 斷詞、詞性標註、命名實體辨識的檢測

雖然本次實驗所包含的工具在各自相關論文中均有提到斷詞成效，但在經過數年的科技發展和資料更替後，我們認為仍需再次驗證其最新結果。所以，本研究將使用以下資料集進行實驗：

- SIGHAN 2003 ~ 2008 年競賽的公開資料集，用以驗證三種工具的斷詞成效。
- SIGHAN 2006 競賽的公開資料集，用以驗證 CKIP 及 MONPA 的命名實體辨識成效。
- 從網路搜集的新聞語料隨機抽出 30 筆文本，經語言專家以人工標註出詞性標註、命名實體辨識等資料，用以驗證詞性標註及命名實體等多工成效。

以往斷詞、詞性標註、命名實體辨識的檢測皆以 Perl 寫成的 conlleval[8] 評分，本研究採用以 python 改寫的 seqeval[9] 套件 (Nakayama, 2018)，並經測試驗證評分標準及結果同 conlleval。

2.2.3 不同斷詞文本對機器學習方法的影響

這部份的實驗資料，是利用爬蟲從網際網路公開網頁搜集約四萬筆新聞文本、四萬筆旅館正負評文本，及 5,500 筆電影正負評文本等三種內容各異的資料。特別注意的是，新聞文本將作為文件分類的資料使用，其中包含六個新聞類型。我們將分別以三種工具對上述正體中文語料進行斷詞，並將結果作為機器學習方法的訓練及測試文本，以藉此實驗輸入不同的斷詞資料是否會影響機器學習分類方法的預測效果。四種機器學習皆是引用 scikit-learn[10] 套件 (Pedregosa et al., 2011) 中內建的方法，包含：

- Naïve Bayes: 使用 ComplementNB() 方法，參數均為預設值。
- Decision Tree: 使用 DecisionTreeClassifier() 函數，參數均為預設值。
- KNN: 使用 KneighborsClassifier()，參數 n_neighbors 設定為 500，其餘皆為預設值。
- SVM: 使用 svm.SVC()，參數 kernel 設定為 linear，gamma 設定為 0.8，C = 1.2，其餘皆為預設值。

3 斷詞效率之實驗結果與討論

3.1 載入套件時間

本實驗在 Jupyter Lab 重啟核心後載入單一套件（標示為 0），隨後以 reload() 重新載入套件兩次（標示為 1 及 2），經紀錄時間後繪製為圖 1 。Jieba 及 JiebaD 為單純 python 套件，並無需先載入其他運行架構，可以從三次測試時間皆相仿得知。但因 JiebaD 要進一步匯入約 4 MB 大小的字典檔，所以在載入套件花費最多時間。CKIP 需要基於 Tensorflow 架構來運作，所以在初次啟動時要先載入 Tensorflow 架構和本身的程式部件等，因此需要花費較多時間。隨後，在 reload 動作進行時，因 Tensorflow 已在運行環境中而僅需載入其自有程式部分，我們推測所需時間主要是花費在載入近 1.8 GB 大小的模型檔上。

[8] CoNLL-2000 shared task, https://www.clips.uantwerpen.be/conll2002/

[9] https://github.com/chakki-works/seqeval

[10] https://scikit-learn.org/

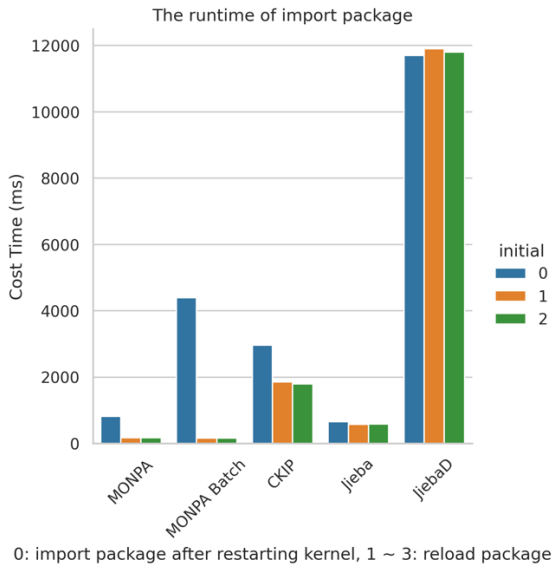

圖 1. 載入斷詞套件所需時間

MONPA 及 MONPA Batch 需要基於 Pytorch 架構來運作，所以同樣在初次執行時要先載入 Pytorch 架構，與自有的套件程式等，因此需要花費較多時間。隨後 reload 動作亦因 Pytorch 已在環境中而僅需重新載入自身套件，所需時間主要是處理近 8.9 MB 大小的模型檔。另外值得注意的是，MONPA Batch 因為需使用 GPU 資源做批次斷詞，所以初次啟動要比 MONPA 預設單線程版本花費更多時間在將模型的參數搬移到 GPU 記憶體中。由此可見，不管是採用深度學習架構或是其他統計式演算法，都需要花費時間載入模型檔或是字典檔。正因如此，模型或資料檔案大小與該斷詞工具的啟動時間高度相關。

3.2 小資料集斷詞時間

準備單句不超過 200 字元長度的正體中文文本，並複製為 1 句到 990 句的不同文本。實驗紀錄各工具處理一篇 1 句到一篇 990 句的文本斷詞，每次執行需要多久時間，不包含載入套件等啟動時間。將時間紀錄繪製成圖 2 。Jieba 及 JiebaD 對 990 句（每句＜200 字元）做文本斷詞的花費時間不會比斷一句的時間多出太多，幾乎呈現水平延伸，表現出快速斷詞的效率。CKIP 對資料多寡的斷詞時間呈現正線性，但增長斜率不大，啟動後預設以多線程執行斷詞任務，表現出穩定的斷詞效率。

MONPA 預設單線程斷詞方法花費時間最多，990 句就要多於 60 秒的執行時間。所以，MONPA Batch 的批次斷詞功能就是改善預設單句斷詞較緩慢的缺點，幾乎貼近 Jieba 的快速效率表現。

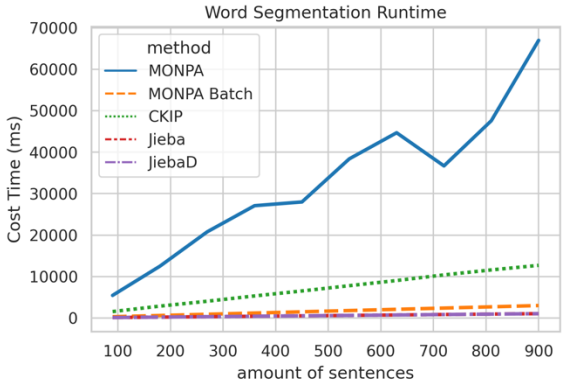

圖 2. 小資料集斷詞耗費時間

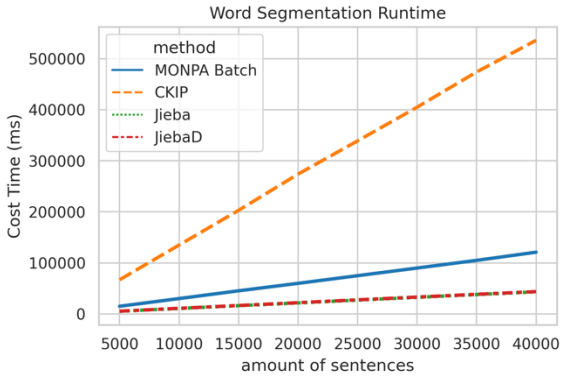

圖 3. 大資料集斷詞耗費時間

3.3 大資料集斷詞時間

本部分實驗將每篇文本包含的句子數增加到 5,000 ~ 40,000 的規模，之後紀錄各工具處理各篇的文本斷詞所需執行時間，同樣的也不包含載入套件等啟動時間。另外，基於前述實驗結果，本部分將排除速度最慢的 MONPA 單線程法。實驗時間紀錄可見圖 3 。Jieba 及 JiebaD 依然是具有最高效率的斷詞工具，CKIP 仍呈現線性增長，但在這個數量規模下，其所花費的時間將明顯高於其他工具。最後，MONPA Batch 的批次斷詞效率表現良好，與 Jieba 的差距不大。整體來說，我們可以得到以下結論：無論小資料或大資料集，在不考慮斷詞正確與否的前提下，Jieba 確實是最快速的斷詞工具。另外，若有 GPU 設備，MONPA 工具可得到大幅度的速度提升。

4 斷詞效能之實驗結果與討論

此部分的實驗針對三種系統的斷詞、詞性標註、命名實體辨識進行檢測，除了考量速度以外，斷詞工具的正確率更是值得關注的指標。因三種工具皆已釋出多年，亦經多次改版，效能應該有所不同。在以下各節中，我們將重新驗證各工具於常用資料集之表現，並討論與分析其結果。

4.1 斷詞驗證：SIGHAN 競賽公開資料集

本部分實驗應用了 SIGHAN 2003 ～ 2008 年的競賽資料集。在三種工具的預設安裝狀態，並且未使用上述搜集的訓練資料集重新訓練的條件下，實驗已載入正體中文字典檔的 JiebaD、CKIP 及 MONPA 對資料集之斷詞成果。我們採用 Precision (P)、 Recall (R) 以及 F1-score (F) 等指標來評估，也就是一個 token 左右兩方的詞界 (boundary) 與標準答案一樣時，視為斷詞正確。

從表 1 可以看出，使用簡體中文語料與 HMM 模型所訓練出來的 Jieba 套件，雖匯入正體中文字典檔，其斷詞效果依然沒有太大提升。而以 Chinese Gigaword 5 (Central News Agency, CNA 部分)、Wikipedia (2019-05-20 pages-articles dump，中文部分)、中央研究院漢語平衡語料庫 (ASBC 4.0) 及 OntoNotes 5.0 (中文部分) 等超過兩千兩百萬句正體中文語句當作訓練語料[11]所開發出的 CKIP ，確實能在 SIGHAN 資料集取得非常好的成績。另一方面，MONPA 雖僅以約十萬句正體中文新聞語料訓練出的套件，應用於 SIGHAN 的資料也有不錯的表現。這也顯示出，使用深度學習方法，搭配足夠大量的資料，能夠獲得令人滿意的訓練結果。因此，建構一個大量且同時含有中文斷詞、詞性標註、以及命名實體資訊的語料庫，是現今中文自然語言處理工具不可或缺的資源，同時也是產學界必須面臨的挑戰。

4.2 命名實體辨識驗證：SIGHAN 2006 競賽公開資料集

命名實體辨識實驗是採用 SIGHAN 2006 年競賽的測試資料集，且斷詞後的命名實體辨識結果，需要同時具備詞界與專有名詞的類型都正確，才會視為辨識成功。Jieba 預設套件沒有命名實體辨識功能，故無法包含於此實驗中。本部分實驗的結果可見表 2。綜合來說，此實驗所包含的兩個工具均有很大的進步空間，我們認為這可能與訓練語料中的命名實體定義標準有關，故進行了接下來的實驗。

System		F_1-Score (%)			
		2003	*2005*	*2006*	*2008*
AS	**Monpa**	94.24	92.33	92.40	93.14
	CKIP	**98.22**	**97.68**	**98.06**	**97.90**
	JiebaD	76.52	73.87	74.32	74.97
City U	**Monpa**	89.10	88.85	89.72	-
	CKIP	**91.50**	**90.59**	**91.61**	-
	JiebaD	72.85	74.06	75.43	-

表 1. 各工具於 Academia Sinica (AS) 與 City University (City U) 資料集之斷詞效能結果

System	F_1-Score (%)			
	LOC	*ORG*	*PER*	*Overall*
MONPA	**74.04**	35.34	79.80	66.94
CKIP	69.75	**37.13**	**88.60**	**67.02**

表 2. 各工具於 SIGHAN 2006 資料集的命名實體辨識效能

System	F_1-Score (%)			
	LOC	*ORG*	*PER*	*Overall*
MONPA	**83.73**	**70.14**	**95.53**	**88.28**
CKIP	79.37	63.38	92.93	79.38

表 3. 各工具於隨機抽選新聞文本的命名實體辨識效能

4.3 綜合驗證：隨機抽選新聞文本

鑑於 CKIP 及 MONPA 對 SIGHAN 2006 資料集的命名實體辨識驗證結果不甚完美，我們另外從網路搜集了 30 則公開的臺灣新聞語料，並請具備語言學背景的專家進行詞性和命名實體標註後，做為本次實驗的驗證測試資料。表 3 的結果支持前述的推論，也就是 CKIP 和 MONPA 在 SIGHAN 2006 效果差強人意，可能與訓練語料的標註標準差異有關。當使用這兩個工具對正體中文的文本斷詞及進行命名實體辨識時，若是採用台灣用語為

[11] https://github.com/ckiplab/ckiptagger/wiki/Corpora

Corpus/System		P/R/F1			
		Naïve Bayes	Decision Tree	KNN	SVM
News (#Train: 30,000, #Test: 10,000)	**Monpa**	78.43/82.83/**79.22**	63.13/64.20/59.40	77.12/81.40/**77.87**	76.69/79.42/**75.12**
	CKIP	78.16/82.62/78.94	63.08/64.30/59.45	76.43/80.49/76.71	76.58/79.28/74.93
	Jieba	77.72/82.09/78.50	63.84/65.09/60.67	76.26/79.63/76.50	75.97/78.79/74.52
	JiebaD	77.74/82.07/78.50	64.45/65.29/**60.94**	76.3079.65/76.53	75.92/78.72/74.45
Hotel Review (#Train: 30,000, #Test: 10,000)	**Monpa**	87.16/85.12/85.78	83.14/83.05/**83.09**	85.98/80.60/**81.60**	88.69/88.73/88.71
	CKIP	87.38/85.15/85.84	82.52/82.30/82.40	85.75/79.80/80.81	89.08/89.13/**89.10**
	Jieba	88.00/85.70/**86.42**	81.36/81.32/81.34	85.77/79.98/80.98	88.91/88.95/88.93
	JiebaD	88.00/85.70/**86.42**	81.25/81.16/81.20	85.85/80.05/81.06	88.91/88.95/88.93
Movie Review (#Train: 5000, #Test: 500)	**Monpa**	82.90/82.83/**82.70**	67.40/67.42/67.39	78.90/78.84/**78.86**	89.19/89.23/**89.20**
	CKIP	82.24/82.21/82.10	69.13/69.14/**69.10**	78.15/78.00/78.03	88.78/88.81/88.79
	Jieba	78.36/78.09/77.87	66.27/66.28/66.28	73.21/73.24/73.20	80.98/80.98/80.98
	JiebaD	78.44/78.19/77.97	63.29/63.30/63.29	73.01/73.04/73.00	81.18/81.17/81.18

表 4. 各工具於三種資料集之斷詞結果用在機器學習分類任務之效能評估結果，粗體字代表在各資料集中表現最佳的方法之 F1 分數。

標準來進行評估，成效將會大幅提升，同時也增進了實用性。

4.4 不同斷詞對機器學習方法的影響

雖然斷詞結果可由標準答案來驗證其成效優劣，但另一方面來看，將斷詞結果當作機器學習分類任務的訓練文本時，不同的斷詞結果可能會影響預測效果。一個好的斷詞工具，應該要能夠產出優良的標註來輔助後續機器學習的任務。因此，我們從網際網路公開網頁搜集共四萬筆，含六種新聞類別的文本資料 (News)，與四萬筆旅店正負評文本資料 (Hotel Review) ，及 5,500 筆電影正負評文本資料 (Movie Review) 。將這些實驗語料分別以三種工具斷詞，並篩選出詞性標註為動詞、名詞、副詞、形容詞、命名實體 (LOC, ORG, PER) 的詞彙組合作為機器學習模型的訓練文本及測試文本，採用前述機器學習的參數進行實驗，結果如表 4 所示。其中可發現，對正體中文斷詞成效較優的 CKIP 與 MONPA 工具，於三種資料集之斷詞結果，同樣也對機器學習分類任務的表現有所助益。

5 結論與未來展望

本研究透過多方面的實驗，評估 Jieba、CKIP、MONPA 等三種在正體中文自然語言研究社群常用的斷詞器，以期找出最適用於大型資料集的斷詞、詞性標註及命名實體辨識的多功能研究工具。在實驗過程中，我們觀察到 MONPA 在 0.3 版本以後，採用 Huggingface[12]工具所提供的預訓練模型資料庫中的 ALBERT[13] (Lan et al., 2020) 模型，替換了初版的 Bi-LSTM 網路後，對比前版在 SIGHAN 的斷詞成效雖略低 0.002 左右，但模型檔案大小也從 55.1 MB 大幅縮減到 8.9 MB，達到以 pip 直接安裝，不須再額外下載模型檔或是字典檔的便利性。為了改善預設單線程斷詞的速率表現，支援利用 GPU 施行批次斷詞，從而大幅縮減巨量中文資料的斷詞時間；而斷詞、詞性標註、命名實體辨識

[12] https://github.com/huggingface/transformers

[13] https://huggingface.co/albert-base-v1

等功能皆可有效輔助中文自然語言處理的後續相關任務提升其表現。

未來，我們期待 MONPA 能進一步加大訓練語料，例如將原先的 10 萬句新聞語料擴展至其他語境的中文語料，甚至是維基百科正體中文版全部資料等，並以此更新其語言模型。基於這樣的巨量資料，可訓練出最貼近語文使用現況的斷詞模型；另外，亦可對主程式作進一步的最佳化，如改善單線程斷詞運行效能或者應用最新深度學習工具進行模型加速等。我們相信，所有正體中文自然語言處理領域的專家、學者、工作者們，都能夠受益於此項研究成果。

致謝

本研究承蒙國家科學及技術委員會計畫 MOST111-2221-E-038-025 之補助，並感謝臺北醫學大學大數據科技及管理研究所研究生陳彥銘及林亨優協助開發爬蟲搜集旅館評論語料及電影評論語料。

References

AFNLP. (2003, July). *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*. https://aclanthology.org/W03-1700

AFNLP. (2005). *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*. https://aclanthology.org/I05-3000

AFNLP. (2008). *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*. https://aclanthology.org/I08-4000

Baum, L. E., Petrie, T., Soules, G. W., & Weiss, N. (1970). A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *Annals of Mathematical Statistics*, *41*, 164–171.

Hsieh, Y.-L., Chang, Y.-C., Huang, Y.-J., Yeh, S.-H., Chen, C.-H., & Hsu, W.-L. (2017). MONPA: Multi-objective Named-entity and Part-of-speech Annotator for Chinese using Recurrent Neural Network. *IJCNLP*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *ArXiv:1909.11942 [Cs]*. http://arxiv.org/abs/1909.11942

Li, P.-H., Fu, T.-J., & Ma, W.-Y. (2020). *Why Attention? Analyze BiLSTM Deficiency and Its Remedies in the Case of NER* (arXiv:1908.11046). arXiv. http://arxiv.org/abs/1908.11046

Nakayama, H. (2018). *seqeval: A Python framework for sequence labeling evaluation*. https://github.com/chakki-works/seqeval

Ng, H. T., & Kwong, O. O. (2006). Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing. *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.