應用多任務序列標記模型於零樣本跨語言網頁模板移除之研究

吳昱豪
資訊工程學系, 國立中央大學
yuhao8888@gmail.com

張嘉惠
資訊工程學系, 國立中央大學
chia@csie.ncu.edu.tw

摘要—網頁雖然資源豐富，但通常與廣告、橫幅、導覽列、版權等模板交織在一起，不利於後續資訊擷取應用。在本文中，我們研究了從輸入網頁中擷取主要內容並去除無關資訊的問題。常見的解決方案是將每個網頁區塊分類爲模板（噪音）或主要內容。**BoilerNet** 等最先進的方法使用神經序列標記在 **CleanEval EN** 資料集中取得了令人印象深刻的分數。在本文中，我們提出了一個基於輔助任務的多任務學習框架：節點路徑深度預測。此外，我們使用多語言 **BERT** 進行文字內容表示來處理任意語言網頁。實驗表明，多任務學習框架在 **CleanEval EN** 資料集上的效能優於 **BoilerNet**。其次，基於多語言 **BERT** 的預訓練文字表示法，雖然在 **EN** 測試集上的性能相近；然而在三種語言（中文、日文和泰文）的零樣本實驗有相當大的提昇，這表明在一個模型中提供跨語言支持的可能性。

***Index Terms*—boilerplate removal** (模板移除), **multi-task learning** (多任務學習), **cross-lingual model** (跨語言模型), **zero-shot learning** (零樣本學習)

I. 緒論

不斷增加的網絡資源已成爲當今廣告的主要市場。因此，網頁不僅包含其主要內容，還包含廣告、導航欄、連結列表、相關文章或橫幅等組件。目前大部分網頁爲了注重點擊率和美觀度，主要文本與上述其他不重要的資訊混合在一起，這增加了資訊檢索或其他應用的難度。圖 1 是自由時報網站的新聞範例，我們用綠色和紅色框標記了主要內容和模板兩種類型的區塊。紅框標出的區域爲模板，通常由延伸閱讀、熱門新聞、廣告等多種連結組成。我們所關心的部分是圖中的綠色區塊，也就是網頁中的主要內文。

擷取網頁的主要內文或從網頁中刪除不重要資訊的任務稱爲模板移除。和爲每個網站學習獨立 wrapper 的網頁資料擷取系統相比，模板移除旨在訓練爲所有網站擷取主要內文的單一模型。舉例來說，對於新聞爬蟲或活動擷取應用，我們需要監控不同網站上發布的最新活動列表，以收集新公告 URL，並執行下載以獲取網頁。由於活動消息嵌入在網頁中，因此刪除網頁模板以擷取主要內文對於活動擷取或精確資訊檢索等下游任務非常重要。另外，去除網頁模板對於網頁廣告也是必不可少的，因爲廣告任務是爲每個頁面匹配相關的廣告。通過模板移除，我們可以根據頁面的主要內容給出更精確的廣告，就像精確的資訊檢索一樣。

圖 1. 台灣自由時報的示例網頁，其中紅色框架中的文本區塊是模板，而綠色框架中的文本區塊是主要內容。

模板移除任務已經被研究了十多年。典型的方法爲基於人工定義特徵的監督式機器學習問題，例如基於文字的特徵 [1]、基於 DOM 樹的特徵、或是基於視

覺的特徵 [2]。先前方法的兩個缺點是難以對與廣告交錯的內容進行局部性建模。在 WWW2020，Leonhardt 等人 [3] 提出了稱爲 BoilerNet 的神經序列標記模型 (Neural Sequence Labeling Model)，該模型將問題建模爲序列標記任務，以消除高計算成本的特徵。

然而，BoilerNet 只考慮單一語言的網頁，在現實中不足以處理全球網頁。此外，爲另一種語言準備訓練資料通常是最耗時的過程。例如，CleanEval 英文資料集包含 741 個（57 個用於開發，684 個用於測試）頁面，而 CleanEval 中文資料集僅包含 50 個頁面。這裡的問題是我們是否可以訓練一個適用於不同語言的模型，而無需標記額外的依賴於語言的訓練資料。因此，我們需要考慮替代方法。例如，是否可以在不增加訓練資料的情況下訓練一種適用於所有語言的模型？換句話說，我們的最終目標是構建零樣本跨語言模板移除模型。

爲了設計用於模板移除的零樣本跨語言模型，我們使用 BoilerNet 作爲基本結構，並探索如何基於標籤路徑和文字序列輸入更好地表示每個文本區塊以進行模板預測。對於標籤路徑輸入，我們提出了標籤路徑深度預測輔助任務，以獲得更好的標籤路徑表示法，並通過多任務學習爲模板預測擷取有用的標籤特徵。對於詞序列輸入，我們用多語言 BERT 句子嵌入替換 BoilerNet 中使用的文字向量，以學習用於模板預測的跨語言文本內容表示。總結來說，本文的貢獻可以概括爲兩個方面。第一，我們透過替換了內文表示向量使得模型在跨語言的能力方面的平均效能從 0.527 提升至 0.766；第二，除了替換內文向量外，我們也使用提出的標籤深度預測任務作爲輔助，在 CleanEval 資料集上從 0.771 提升到 0.798，是超越了 BoilerNet 所達到的 0.774，在跨語言實驗上也再提升了 6.1% 的 F1 分數。

II. 相關研究

模板去除的任務也稱爲主要內容擷取問題（因爲目標是去除模板並擷取頁面中的主要內文節點），屬於網絡資料擷取的廣泛研究領域。

對於模板移除任務，Baroni 等人提出了一個關於清理任意網頁主題的共享任務和競爭評估，一個名爲“CleanEval”[4] 的公開資料集。資料集包括英文訓練集、開發集、測試集以及中文測試集。CleanEval 被廣泛用於測量網頁上的各種模板移除方法。

Spousta 等人 [5] 最早將模板移除任務定義爲序列標記問題，並採用了條件隨機場 (CRF) [6] 對 CleanEval 資料集進行實驗。Kohlschütter 等人 [1] 在 2010 年提出了 Boilerpipe，旨在避免依賴網域的上下文特徵，專注於更簡單和更高級別的文本特徵，以節省成本高昂的特徵，如視覺特徵。Vogels 等人 [7] 則提出了 Web2Text，它使用壓縮 (Collapse) DOM(Document Object Model) 架構將網頁分割成區塊。對於每個文本區塊，We b2Text 定義了來自單個和相鄰葉節點的 128 個區塊特徵和 25 個邊特徵作爲輸入特徵。他們使用兩個 5 層 CNN（即兩個並行 CNN），kernel 大小爲 (1, 1, 3, 3, 3)，並使用隱馬爾可夫模型 (Hidden Markov Model) 擷取用於模板預測的特徵資訊。

當今效能最佳的方法是 Leonhardt 等人在 2020 年提出的 BoilerNet [3]。與之前介紹的方法相比，BoilerNet 並沒有使用大量人爲定義的特徵。取而代之的是，每個網頁都基於 DOM 樹葉節點分割來劃分爲文本區塊。BoilerNet 使用頻率最高的 50 個 HTML 標籤和 1000 個詞，做爲輸入標籤向量和詞向量。對於模型設計，使用兩層雙向 LSTM 來擷取文本節點之間的關係。實驗結果顯示 CleanEval EN 資料集上的 micro F1 爲 0.83。

表 I 展示了上面介紹的四種方法的對比。Victor 和 Boilerpipe 都使用 HTML 標籤作爲基本單位，而 Web2Text 和 BoilerNet 則使用 DOM 葉節點分割頁面。在模型設計上，BoilerPipe 使用 SVM（支持向量機），而 Victor 是第一個使用 CRF（條件隨機場）來解決問題的分類器。Web2Text 和 BoilerNet 後來提出了使用 CNN 和 LSTM 結構的神經網絡模型，在預測層使用 HMM（隱馬爾可夫模型）和 MLP。在輸入特徵表示方面，像 SVM 和 CRF 這樣的傳統機器學習方法需要進行特徵工程，採用例如“節點是否爲 container ?”、“節點是否爲連結？”、詞/句子計數（在 Victor 中提出）。Boilerpipe 不僅使用平均單詞/句子長度，還使用高級文本特徵，如文本區塊的文字密度，以避免捕捉網站的特定格式和頁面渲染成本。Web2text 使用了區塊特徵和邊特徵。前者包括人爲定義特徵，而後者則捕捉區塊之間的關係，例如，他們將“樹距離”定義爲從節點到其第一個共同祖先的距離和，並使用“樹距

離”定義多個二元特徵。最後，BoilerNet 則選擇出現頻率最高的標籤和詞，並通過一個標籤向量和一個詞向量來表示每個文本段。

表 I
COMPARISON OF BOILERPLATE REMOVAL METHODS

Method	Delimiter	Model	Features
Victor	HTML Tag	CRF	HTML, Text
Boilerpipe	HTML Tag	SVM	Text, Density
Web2Text	CDOM Leaf	CNN+HMM	Block, Edge
BoilerNet	DOM Leaf	LSTM+MLP	Tag/Word vector

III. 模型設計

與 BoilerNet 類似，我們採用神經序列標記模型來解決模板去除問題。我們使用 BeautifulSoup 函式庫基於 DOM 樹結構將每個輸入網頁劃分爲文本區塊做爲輸入，再根據相應的 HTML 標籤路徑和文本內容將每個文本區塊分類爲模板或主要內容。

令 $\mathbf{x} = \{x_1, x_2, x_3, ..., x_n\}$ 表示一個具有 n 個葉節點的網頁，其中每個葉節點 $x_i = \{p_i, s_i\}$ 由標籤路徑 $p_i = (t_1^i, t_2^i, ..., t_{|p_i|}^i)$ 和詞序列 $s_i = (w_1^i, w_2^i, \ldots, w_{|s_i|}^i)$，其中 $|p_i|$ 和 $|s_i|$ 分別是標籤路徑 p_i 和單詞序列 s_i 的長度。每個葉節點的輸出爲一個標籤 $y_i \in \{0, 1\}$，指示文本區塊是主要內文還是模板，我們的目標是預測 $\mathbf{x}$ 中所有文本區塊的標籤。

圖 2 中提出的模型包括以下內容組件：輸入表示層（包括一個標籤路徑編碼器和一個詞序列編碼器），用於擷取關係文本區塊的貝氏 BiLSTM 層，以及兩個預測任務（包括用於主要任務的模板分類器和用於主要任務的節點深度預測器）輔助任務）。

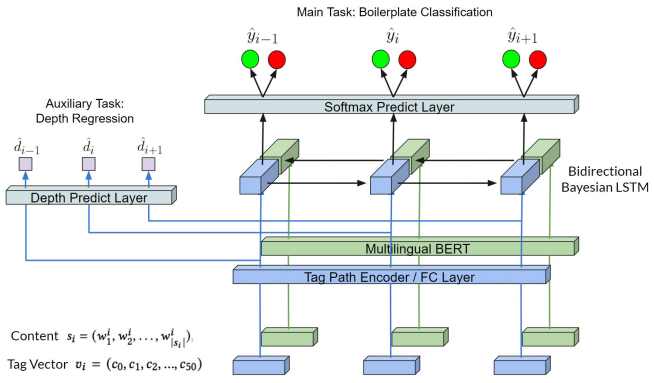

圖 2. MultiBoilerNet 模型

A. 輸入表示

由於詞向量表示依賴於語言，因此我們用多語言 BERT [8] 預訓練嵌入替換了 BoilerNet 中使用的詞向量。根據 BERT 輸入格式，我們在詞序列 s_i 的開頭和結尾分別加入 [CLS] 和 [SEP] 兩個標記，如方程式 (1) 所示。[CLS] 標記的輸出向量即爲文本內容表示。

$$TC_i = BERT([CLS], w_1^i, w_2^i, \ldots, w_{|s_i|}^i, [SEP]) \quad (1)$$

對於標籤向量表示，我們遵循 BoilerNet [3] 選擇頻率最高的前 50 個 HTML 標籤，並將每個標籤與一個維度相關聯。然後我們將每個標籤轉換成一個 51 維的標籤向量，$v_i = (c_0, c_1, c_2, ..., c_{50})$，其中 c_k $(1 \le k \le 50)$ 表示第 k 個標籤在標籤路徑 p_i 中的出現次數，c_0 爲不在前 50 個標籤中的標籤出現次數。最後我們採用全連接層將標籤向量轉換爲較低維的密集向量。

B. Variational LSTM Dropout

給定來自標籤路徑和文本內容表示的隱藏層，我們將它們連接起來並使用 BiLSTM 來學習文本區塊序列中的上下文關係。LSTM 是由 Hochreiter 等人提出的 [9] 克服了訓練循環神經網絡 (RNN) 的時間序列資料的困難。然而，Gal 等人 [10] 指出 Dropout 技術在 LSTM 上效果不佳，因爲 Dropout 技術會爲每個輸入樣本生成新的 Dropout Mask，而不管它來自哪個時間序列。因此他們提出了 Variational LSTM，它爲每個輸入序列生成一個 Dropout Mask，並在每個序列的不同時間輸入保持相同，以確保網絡隱藏狀態中的元素將在整個序列中持續存在。

C. 訓練目標

如圖 2所示，除了主要的模板預測之外，我們還設計了輔助任務來指導模型訓練。

1) 模板分類器: 對於主要任務，我們採用分類交叉熵 (Cross Entropy) 作爲損失函數，以最小化每個文本區塊的主要內容/模板的預測誤差，如 Eq. (2) 中所定義。

$$\text{Main Loss} \quad L_{ce} = -\sum_{i=0}^{n} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \quad (2)$$

2) 輔助任務: 對於標籤向量編碼器，我們提出了一個基於路徑深度預測的輔助任務來學習標籤路徑表示。我們將任務建模爲回歸 (Regression) 問題，以預測每個文本區塊的標籤深度（即 $|p_i|$）。對於回歸任務，我們採用均方誤差 (MSE)，L_{depth} 如 Eq. (3) 所示，其中 $\hat{d}_i$ 是第 i 個節點的深度預測。假設我們在一個網頁中有一個葉子節點，它的完整標籤路徑是：[html, body, div, div, p]，那麼這個節點的深度應該是 5（如果是根的深度的話）tree 定義爲 1)，在損失函數中我們採用均方誤差 (MSE)，L_{depth} 定義爲 Eq. (3)：

$$\text{Depth Loss} \quad L_{dep} = \sum_{i=0}^{n} MSE(|p_i|, \hat{d}_i) \tag{3}$$

模型通過最小化輔助任務損失函數來學習預測網頁標籤的表示法，達到提升主任務的效果。

我們將回歸損失函數與主要損失 L_{ce} 結合作爲整體多任務損失 L_{total}（參照 Eq. (4)），其中 α 是權重因子的輔助損失，用於在多任務學習框架下調整輔助任務的重要性。

$$\text{Total Loss} \quad L_{total} = (1-\alpha)L_{ce} + \alpha L_{dep} \tag{4}$$

IV. 實驗

爲了評估所提出的多任務模型的性能，我們採用 BoilerNet 清理的 CleanEval [4] 資料集作爲主要資料集（如表 II 所示）。此外，我們還收集並標註了來自日本和泰國的新聞網頁，作爲評估模型跨語言能力的資料集。CleanEval EN 資料集有 58 個訓練網頁和 684 個測試網頁。訓練集和測試集的平均葉節點數約爲 200，標籤分佈約爲 1:1。請注意，EN 資料集包含來自多個領域的網頁，包括新聞、部落格、搜索結果等，而 ZH 資料集僅包含由多數部落格及少數的新聞的網頁組成共 50 個網頁。

表 II
CleanEval 的資料統計（前兩列）和兩個新資料集（日文和泰文）

資料集	類型	頁數	平均節點數	網域	內容%
EN	開發	58	232.7	Multiple	47.3%
	測試	684	198.6	Multiple	57.2%
ZH	開發	50	219.5	部落格、新聞	38.3%
JP	測試	60	355.4	新聞	2.1%
Thai	測試	60	172.3	新聞	11.1%

除了 CleanEval 資料集，我們還準備了日文和泰文網頁，並手動將每個文本區塊標記爲模板或主要內容。日文網頁從朝日、讀賣、產經、時事、西日本、東京-np 6 個網站收集；對於泰文網頁，則是從 K@POOK、sanook、CH3Plus、Khaosod、Daradaily 和 Thairath 網站抓取的。新收集的頁面和 CleanEval 資料集的主要區別在於模板與主要內容的比例爲 9:1，尤其是在日文新聞頁面中，只有 2.1% 的文本區塊是主要內容，顯示出過多的模板資訊。

A. 模型和設置

在接下來的實驗中，我們使用 BoilerNet 作爲基準，並與基於多語言 BERT 表示的建議模型進行比較。預設模型（標記爲 MultiBoilerNet）是：使用文本內容表示的預訓練 BERT，結合標籤路徑表示的標籤向量編碼器，以及作爲多任務學習框架輔助任務的深度預測。儘管 Finetune 的 BERT [8] 在許多 NLP 下游任務中表現良好，但序列標記的參數計算記憶體需求太大，無法一次性輸入所有文本區塊。透過 Finetune，資料輸入一次最多允許 10 個文本區塊。因此，預設模型僅將預訓練的 BERT 句子嵌入作爲輸入，而不進行 Finetune，並批量處理給定頁面的所有文本區塊。

我們從開發集中保留 5 個網頁用於驗證以決定以下參數：驗證後我們的最終模型包含兩個雙向 LSTM 層，每個層有 256 個隱藏單元。輸入的標籤向量被投影到 256 維空間作爲標籤表示。我們爲貝氏 LSTM 模型設置了 Dropout 率 0.1 和 LSTM Dropout 率 0.01。我們將每個模型訓練 20 個 epoch，alpha 設定 0.5，最後根據驗證集上的平均 Macro F1 分數選擇最佳模型。

B. *CleanEval* 原始任務

第一個實驗是 CleanEval 的原始任務，其中模型在 CleanEval En 開發集上進行訓練。表III 和表IV 分別顯示了 CleanEval EN 測試集上的 Micro F1 和 Macro F1。除了預設的 MultiBoilerNet 模型，我們還考慮了兩種可能的組合：Finetune BERT 和 MultiBoilerNet 與使用一般 BiLSTM 的模型。

正如我們所見，Finetune 的 BERT 模型的性能最差。與 BoilerNet 相比，平均 Micro 和 Macro F1 分數分別顯著下降了 9% 和 10.7%。我們認爲此模型性能不佳是因爲我們爲了適用序列標記會使用大量記憶體的需求，無法保證所有文本區塊能夠一次性輸入，而將文本區塊限制在每批擷取 10 個。另一方面，預設的 MultiBoilerNet 模型使用一般或貝氏 BiLSTM 提高了

Macro F1 分數的性能（2.4 到 2.7%），儘管 Micro F1 分數的差異很小（0.5 到 0.9%）。

表 III
CleanEval Original Task: EN Test Set (Micro F1)

Model/Performance	Noise F1	Content F1	Avg. F1	Diff %
Boilerpipe [1]	0.71	0.71	0.71	-14.0
Web2Text [7]	0.79	0.86	0.83	-2.0
BoilerNet [3]	0.82	0.87	0.85	–
Finetune BERT	0.730	0.783	0.756	-9.0
MB-BiLSTM	0.831	0.870	0.851	0.5
MB-BayBiLSTM	0.831	0.878	0.855	0.9

表 IV
CleanEval Original Task: EN Test Set (Macro F1)

Model/Performance	Noise F1	Content F1	Avg. F1	Diff %
BoilerNet [3]	0.730	0.818	0.774	–
Finetune BERT	0.617	0.718	0.667	-10.7
MB-BiLSTM	0.767	0.836	0.801	2.7
MB-BayBiLSTM	0.760	0.836	0.798	2.4

C. ZeroShot from EN to ZH, JP and Thai

由於我們的最終目標是構建零樣本跨語言模板移除模型，來處理全球網頁。同時我們不希望花費大量時間來準備不同語言的訓練資料，因此在第二個實驗中，我們應用了由 CleanEval EN 構建的四個模型，並在三個資料集上對其進行了測試：CleanEval ZH 資料集、JP 和 Thai。Macro F1 分數的結果列在表 V中。在每個表的最後一行，我們還展示了每個測試集的 5 倍交叉驗證性能的平均 Macro F1 作爲參考。

首先，在表 V中，我們發現 BoilerNet [3] 在識別內容文本區塊的能力大大降低（Macro F1 僅爲 0.098）因爲內容特徵是從英語資料集中擷取的，即語言相關。因此，BoilerNet 模型傾向於預測“Noise”標籤而不是“Content”標籤，導致平均 Macro F1 分數爲 0.404。相比之下，帶有 Finetune BERT 的 MultiBoilerNet，由於使用了多語言 BERT，該模型仍然可以理解未經訓練/新語言的文本。與 BoilerNet 相比，在 F1 分數上的性能平均提高了 30%。預設 MultiBoilerNet 模型（基於預訓練的 BERT 和一批完整的文本區塊）在 Zh 資料集上對 BoilerNet 的改進更令人印象深刻：Macro F1 分數提高了 44%。特別是，貝氏 LSTM 在 ZH 測試資料集上超過了 5 倍交叉驗證的性能（0.862），達到了 0.869。結果表明，如果訓練資料是完整、乾淨和多領域的，我們就有機會獲得比原始語言更高的性能。

表 V
ZeroShot Experiments (Train on EN, Macro F1)

Model/Performance	ZH Avg. F1	JP Avg. F1	Thai Avg. F1	Diff %
BoilerNet [3]	0.404	0.490	0.688	–
Finetune BERT	0.704	0.661	0.632	13.8
MB-BiLSTM	0.844	0.816	0.771	28.3
MultiBoilerNet	0.869	0.816	0.779	29.4
5 Fold CV	0.862	0.917	0.814	–

其次，對於日文和泰文的零樣本測試，我們也有類似的結果。正如上面表 II中提到的，主要內容標籤的百分比極低：日文僅爲 2.1%，泰文爲 11.1%。因此，預測“Noise”標籤的趨勢更強，導致“Noise”標籤 (0.980) 上的 Macro F1 較高，但“Content”標籤 (0) 上的 Macro F1 較低。BoilerNet 的平均 Macro F1 分數爲 0.490。對於 MultiBoilerNet，Finetune BERT 的平均 Macro F1 分數爲 0.661，而 MultiBoilerNet 的分數達到 0.832。

值得注意的是，泰文資料集比 ZH 和 JP 資料集包含更多的英語單詞。用英文編寫的這一小部分爲 BoilerNet [3] 的“Content”標籤提供了 0.419 的 Macro F1 分數，優於 Finetune BERT。總而言之，預設的 MultiBoilerNet 在所有零樣本實驗中都取得了最好的分數，結果非常接近 5 Fold 交叉驗證的結果。

D. 輔助任務的作用

爲了理解多任務學習框架的效果，我們訓練了一個沒有路徑深度預測輔助任務的模型，即使用主要損失函數作爲我們的訓練目標，並在四個資料集上測試性能。每個資料集的 Macro F1 得分如圖 3 所示。我們可以看到，基於深度預測的輔助任務在所有測試資料集中一致地將模板移除的性能提高了大約 2~7%（平均 6%）。

E. 消融實驗

圖 4 展示了模型中每個特徵的切除研究，從左到右分別是“Vector+BERT”，這是 MultiBoiler 的預設模型，達到最高性能；“Vector Only”和“BERT Only”

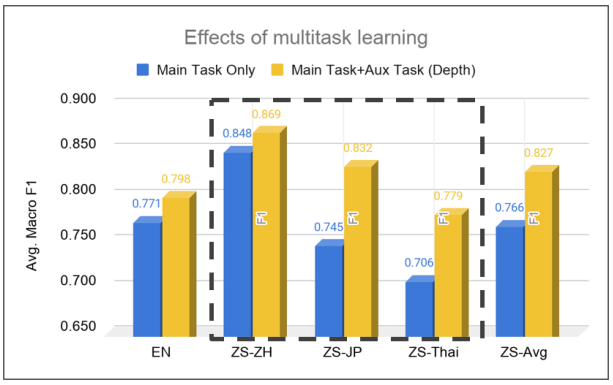

圖 3. Effect of the auxiliary task based on depth regression

是只使用一個特徵的模型，從圖中可知 BERT 在整體模型的貢獻上佔比較大。

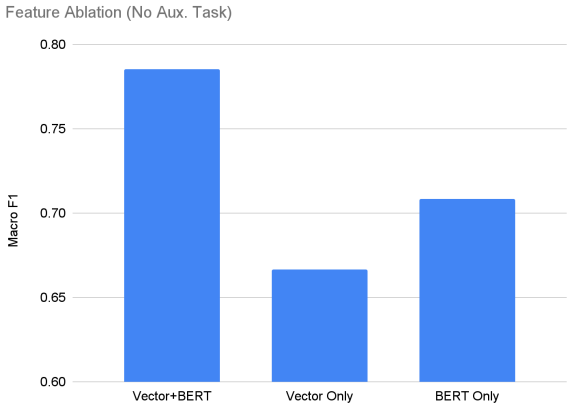

圖 4. Feature Ablation w/o Aux. Task

V. 結論

現實世界的網際網路環境是多語言的，我們需要一個模型來擷取主要內容並刪除任何頁面中的模板以進行精確的資訊檢索或廣告投放。在本文中，我們在多任務學習框架下提出了一種用於多語言模板移除的神經序列標記模型。我們用預訓練的多語言編碼器 BERT 替換了 BoilerNet 使用的內文向量，然後引入了預測當前文本區塊在 DOM 樹中標籤路徑的深度輔助任務，以學習更好的標籤路徑表示來提高模型性能，

在實驗方面，Finetune BERT 參數對資源的需求太大，因此當我們一次只能處理 10 個文本區塊時，性能受到限制。通過批量加載頁面的所有文本節點（無需 Finetune），預設 MultiBoilerNet 在 EN 測試集上的表現優於 BoilerNet 0.9% Micro F1 和 2.4% Macro F1。我們對中文、日文和泰文進行了三個零樣本實驗。在多任務學習的幫助下，平均 Macro F1 分數可以從 2.1 提高到 8.7%。對於未來的工作，我們計劃嘗試域對抗神經網絡來進一步改進 MultiBoilerNet 模型。

參考文獻

[1] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ser. WSDM '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 441—450. [Online]. Available: https://doi.org/10.1145/1718487.1718542

[2] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, "Block-based web search," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 456—463. [Online]. Available: https://doi.org/10.1145/1008992.1009070

[3] J. Leonhardt, A. Anand, and M. Khosla, "Boilerplate removal using a neural sequence labeling model," in *Companion Proceedings of the Web Conference 2020*, ser. WWW '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 226—229. [Online]. Available: https://doi.org/10.1145/3366424.3383547

[4] M. Baroni, F. Chantree, A. Kilgarriff, and S. Sharoff, "Cleaneval: a competition for cleaning web pages," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA), May 2008, p. 6.

[5] M. Spousta, M. Marek, and P. Pecina, "Victor : the web-page cleaning tool," in *The 4th Web as Corpus Workshop (WAC4)-Can we beat Google*. Marrakech, Morocco: European Language Resources Association (ELRA), 2008, pp. 12–17.

[6] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, p. 282—289.

[7] T. Vogels, O.-E. Ganea, and C. Eickhoff, "Web2text: Deep structured boilerplate removal," in *European Conference on Information Retrieval*, Springer. Grenoble, France: Springer, 2018, pp. 167–179.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[10] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 1027—1035.