The 33rd Conference on Computational Linguistics and Speech Processing (ROCLING 2021)
Taoyuan, Taiwan, October 15-16, 2021. The Association for Computational Linguistics and Chinese Language Processing

應用對抗式 Reptile 於家電產品網路評論之研究
Home Appliance Review Research Via Adversarial Reptile

甘岱融 **Tai-Jung Kan**
中央大學資訊工程學系
j7400660@gmail.com

張嘉惠 **Chia-Hui Chang**
中央大學資訊工程學系
chia@csie.ncu.edu.tw

莊秀敏 **Hsiu-Min Chuang**
國防大學理工學院資訊工程學系
showmin1205@gmail.com

摘要

對於生產家電廠品的廠商來說，收集及分析產品在社群平台上被討論及喜好程度是企業長久經營相當重要的部份。本論文探討家電產品的評論分析方法，尤其是提及產品不同面向的優劣，是產品設計改良時很好的指標。在本篇論文中，我們將家電產品評論分析分成三個子任務：產品名稱辨識、意見標的面向種類擷取、以及情緒分類。我們首先以 BERT 爲基礎的模型，做爲三項任務的基本效能。其次在情緒分類的任務中，嘗試針對不同的意見目標面向訓練任務導向的模型。此部分我們結合遷移式學習中的 Reptile 演算法、以及對抗式訓練的概念，組成對抗式 Reptile 演算法。研究結果顯示，在多模型的基礎上，使用了對抗式 Reptile 架構訓練後的模型，其 Macro-F1 達到 70.3%，較原始數值 (68.6%) 有些微提升，統計 p 值爲 0.04，獨立樣本檢定爲顯著差異，顯示遷移式學習有助於情緒分類任務提升其效能。

Abstract

For manufacturers of home appliances, the Studying discussion of products on social media can help manufacturers improve their products. Opinions provided through online reviews can immediately reflect whether the product is accepted by people, and which aspect of the product are most discussed . In this article, we divide the analysis of home appliances into three tasks, including named entity recognition (NER), aspect category extraction (ACE), and aspect category sentiment classification (ACSC). To improve the performance of ACSC, we combine the Reptile algorithm in meta learning with the concept of domain adversarial training to form the concept of the Adversarial Reptile algorithm. We find show that the macro-f1 is improved from 68.6% (BERT fine tuned model) to 70.3% (p-value 0.04).

關鍵字：意見目標情緒分析、意見面向擷取、元學習、遷移式學習

Keywords: aspect-based sentiment analysis, aspect category classification, meta-learning, transfer learning

1 緒論

網路社群媒體發展蓬勃，造成社群平台上眾多不同類型的討論，龐大的討論資料是輿情分析最佳資訊來源。以實體商品爲例，產品製造商除了最基本的廠牌名稱以及產品名稱之外，也需要知道該評論針對該產品的何種面向進行討論。然而對於不同類型的商品，通常會有各自不同的討論面向。例如，對於家電產品來說，功能可以視作一種討論標的面向，但在討論美食時，就不會存在功能這個討論標的。

在本研究中，我們將商品種類限定在「家電產品」，並且將網路評論分析分成廠牌、產品名稱、討論標的、蘊含情緒等四種不同資訊。舉例來言，「不推日立除濕機，雖然耐用但很吵。」，該句話包含的廠牌及產品名稱分別爲日立、除濕機，而對於該產品的評論有兩個面向：「耐用」在目標種類中可以歸類爲品質，屬於正向評價、「很吵」歸類爲音量，屬於負向評論。我們可將上述三種資訊簡單分類成三種不同的任務。首先，針對產品名稱以及廠牌名稱，可以作爲命名實體辨識 (Name Entity Recognition, NER) 進行識別；而對於判定產品的目標種類可以視爲是面向種類擷取 (Aspect Category Extraction, ACE)；情緒辨識則是基於某產品的某目標種類的情緒表，，可以看成基於目標種類的情感分類 (Aspect Category Sentiment Classification, ACSC) 任務。

我們首先使用 BERT 爲基礎的表示法，對命名實體辨識 (NER)、面向類別擷取 (ACE)、基於目標種類的情感分類 (ACSC) 三種任務建構基礎模型、得出這些任務的基準效能。對於 NER 任務我們採用 BERT-BiLSTM-CRF 序

列標記架構，針對廠牌及產品分別得到 94.2% 及 93.6%F1 效能。而在 ACE 及 ACSC 兩個任務上，我們參考了 Sun(Sun et al., 2019b) 等人的做法，在輸入方式改成以輔助句進行分類，在 ACE 任務上得到 68.1% Micro F1，另外在 ACSC 任務上效能僅有 73.6% F1。

爲改善 ACSC 效能，我們分析了不同的意見標的面向的情感分析效能。由於同一字詞在不同意見標的面向有不同情緒的標籤，例如「高」這個字在品質面向是代表正向情緒，但在音量面向上就會是負面情緒。因此在不同的意見標面向，其文字敘述都有一定程度的差異。針對此問題，我們嘗試爲不同的意見標的面向，訓練各自專屬的情緒偵測模型。同時由於某些意見面向的訓練資料過少，因此我們試著藉助其他不同領域的情感分類資料來做遷移式學習 (Transfer Learning)。我們選用適合做少樣本學習的元學習 (Meta Learning) 方法，並採用其中的 Reptile 演算法 (Nichol and Schulman, 2017)，結合對抗式學習 (Ganin et al., 2016) 的概念，提出了對抗式 Reptile(Adversarial Reptile) 的訓練架構。

首先比較使用單模型，無任何遷移式學習訓練的模型效能作爲基礎，觀察使用對抗式 Reptile 的架構在多模型的基礎上訓練後的模型是否能提升效能，在 Macro-F1 由 68.6% 提升至 70.3%，並且統計 p 值爲 0.04，低於 0.05，獨立樣本檢定爲顯著差異。接著比較 Reptile、DANN 兩者與對抗式 Reptile 的消融實驗，發現對抗式 Reptile 在計算上的貢獻主要幾乎都來自於 Reptile 演算法。

雖然目前元學習在諸多領域都有相關的研究，但在自然語言處理上的發展仍較少，因此本論文藉由元學習探討其在 ACSC 任務上的發展也提供了未來若有人想要在自然語言上應用元學習的相關依據。

2 相關研究

2.1 意見分析

線上評論分析是一種意見擷取最主要的應用，其目的在一段話中找出其中我們關注的資訊，例如 Amazon 的書評 (Aashutosh Bhatt, 2015; A. Mounika, 2019)、旅館評論分析 (Walter Kasper, 2011)、航空公司評論 (Ayat Zaki Ahmed, 2020) 等等。意見擷取中最基本的任務即是情感分析 (Sentiment Analysis)，情感分析又可依據情感對象的不同分成文件層級、句子層級、以及意見目標層級三種。文件層級是分析一個文本的情感，句子層級則是判斷每個句子的情感，意見面向情感分析 (Aspect Based Sentiment Analysis, ABSA) 則是針對句子中不同的面向給出相對應的情感分析。

意見面向情感分析依據是否額外參考意見標的詞，又可分爲基於面向的目標詞情感分析 (Targeted-ABSA, TABSA) 以及意見面向類別情感分析 (Aspect Category Sentiment Analysis, ACSA)。當文中沒有具體提及意見目標，而是以隱式表達某件事物的角度時，我們定義意見面向類別 (Aspect Category)，目的在判別出句子中提及的目標面向類別，再對這些類別進行情感辨識，因此稱此爲目標類情感分類問題 (Aspect Category Sentiment Classification, ACSC)。在 TABSA 的任務中，則是給定句中可能會出現的目標詞 (Target Word)，判斷文句對於意見目標的目標種類及其正、負或中立情感意見。TABSA 與 ACSA 兩者最大的差距在於，TABSA 是針對句子中的給定目標詞進行分析，但 ACSA 則是針對整個句子進行分析。

在 TABSA 的任務中，目前最知名的訓練方式是應用 BERT 中對於句子對優異的預測效能，使用輔助句子 (Auxiliary Sentences) 協助預測 (Sun et al., 2019b)，該篇論文將輔助句子依照句子類型及輸出模式，提出四種輔助句子。句子類型分爲句子推論 NLI(Natural Language Inference)、問答 QA(Question-Answer) 兩種：NLI 類型的句子爲虛擬句子，僅包含關鍵字，不具有實際上的語句意義；而 QA 則是問句類型的句子。輸出則分爲二元 (Binary)、多元 (Multiple) 兩種，二元輸出的輔助句子包含所有資訊 (目標詞、目標種類)，任務是要判斷該輔助句子所意涵的資訊是否正確；多元輸出的輔助句子資訊僅包含目標詞及目標種類，目標是希望判別何種情緒。依上述各二的分類可以組合成四種不同的輔助句子：NLI-M、NLI-B、QA-M、QA-B。

2.2 遷移式學習

遷移式學習的概念是將一個領域學習到的知識模型應用到其他不同但相關的問題。當訓練資料不足，需要藉助其他相似的任務來輔助學習時，遷移式學習的模型架構、訓練方式可以做爲模型效能提升的解決方法。在遷移式學習中，通常會分成來源領域及目標領域 (Target Domain)，根據來源領域及目標領域各有無標記資料的情況，有不同的遷移式學習的方法。當來源領域及目標領域都有標記資料時，最常使用的方法有兩種，分別是微調 (Fine-Tuning) 及多任務學習 (Multi-Task)(Caruana, 1998)。微調是將模型在來源領域上訓練完之後，再放到目標領域上進行訓練，做參數的調整，由於概念簡單，所以在各種不同的領域

上都有應用 (Sun et al., 2019a; Nguyen et al., 2020)。而多任務學習的方法就是將不同來源領域及目標領域的資料放在一起學習，在模型中不同領域或任務會對應到不同的輸出，但其中會有部分的參數或網路層是兩個領域共享的，較知名的例子是多語言的同步翻譯 (Huang et al., 2013)。

當目標領域沒有標記資料時，例如來源領域與目標領域資料分佈不同時，最常使用的方法爲領域對抗學習 (Domain-Adversarial Training of Neural Networks, DANN)(Ganin et al., 2016)。雖然缺少目標領域標記資料，但是由於知道來源領域與目標領域兩種資料，相當於多一個自我標記的資訊。因此 DANN 的設計即是再額外訓練一個領域判別器 (Domain Discriminator)，專門用來判別每筆資料是來自什麼領域。從架構上來說，DANN 的模型架構先將輸入資料進行特徵擷取，也就是相當於資料的特徵擷取器 (Feature Extractor)，再利用聯合學習 (Joint Learning) 機制，分別進行原始來源任務、以及領域分類兩種任務的最佳化。DANN 的核心概念是在特徵擷取器的輸出與領域分類任務中間接上梯度反轉層 (Gradient Reversal Layer)，這樣的設計可以迫使特徵擷取器找出其他眞正幫助原始來源任務的特徵，而非是幫助領域判別器結果的特徵，也就是說應用梯度反轉層來確保找出對兩個領域都共有的特徵，使得領域分類器盡可能無法區分資料來源域。

除了上述幾種經典的遷移式學習方法外，近幾年也出現了一個新的分支: 元學習 (Meta Learning)。元學習並不將來源領域與目標領域一起訓練，而是在來源領域上學習出一個學習能力強的模型，再放到目標領域上重新訓練。假設來源領域有多種不同的任務，而目標領域是一種在來源領域並未出現過的任務，元學習的目標即是透過來源領域的不同任務學習到一個能夠適應不同任務的模型，再將這個模型放入目標領域中進行學習，期許能在較少量資料的情況下學習出一個效能強的模型，近期研究也證實元學習對於這種小樣本學習 (Few-shot Learning) 的問題有相當好的成效。簡言之，元學習的核心思想是讓機器「學習如何學習」。

近年來在不同的任務上也出現不少元學習的相關演算法，而絕大多數的元學習算法都可以歸納成雙層優化問題 (Bilevel Optimization)。其包含了兩層迴圈：Inner-loop 及 Outer-loop，分別進行不同目標的優化。在元學習中，模型主要分成個別任務模型 (Base-Learner) 以及元模型 (Meta-Learner) 兩種，而這兩種模型的架構都長得一樣，差別在於優化目標不同。在 Inner-loop 中，首先會對於來源領域的每個任務訓練其個別任務模型，訓練完成後在 Outer-loop 中利用前訓練的模型結果更新元模型的參數，再將原模型的參數重新賦予給每個個別任務模型進行初始化，之後進入下一次迭代，迭代完成的元模型即是該元學習算法得出的結果。多數的元學習算法就是在 Inner-loop 及 Outer-loop 的參數更新或優化方式進行更改。

3 家電評論意見資料集

我們從生活網站論壇「Mobile01」中，與家用電器有關的五個版下載其中的發文及討論串內容，作爲原始文章。並從經濟部網站中蒐集家電產品名稱、廠牌相關種子進行篩選及段落裁切，共擷取出 7,195 個段落，並隨機抽取 2,000 份進行人工標記。人工標記的項目包含「產品名稱」、「廠牌名稱」、「目標種類」、「目標情緒」，其中目標種類分爲功能、品質、外觀、音量、售後、價位、配件、其他，以上八項；情緒則分爲正向、負向、中立三種。而在段落中對於某特定產品的其中一個目標種類敘述標記，則稱爲一個「關係」。

資料標記由兩人進行獨立標記，再評估兩份標記結果的一致性，以確保資料標記的可靠度，在計算一致性時，我們採用 Cohen Kappa(McHugh, 2012) 值進行一致性計算。在我們標記資料中的 Kappa 值計算需要分成兩部分：一是對於標記實體的 Kappa 值計算，此所述實體包含了標記項目中的產品名稱、廠牌名稱；二是對於標記類別的 Kappa 值計算，包含目標種類、目標情緒兩類。通常在 Kappa 值大於 0.6 時便認爲是有較高的一致性，而低於 0.6 則認爲一致性較低。

對於實體類型的 Kappa 值計算，Alex 等人 (Brandsen et al., 2020) 認爲，若是基於字元進行 Kappa 值計算會出現過多的負向標記，很容易高估實際的一致性。而若是基於標記實體進行計算反倒因爲不存在兩人皆未標註的實體資訊，而缺少負向標記的資料，而在計算結果上嚴重低估其實際一致性。因此我們嘗試在實體基礎的算法上添加負向標記的資料。我們根據該實體的平均字元長度以及目前段落長度去預估該段落存在的所有實體數量，並將預估數量減去有標記的實體數量，作爲負向標記的實體資訊。按照上述方法算出的產品名稱及廠牌名稱標記 Kappa 值分別爲 0.959 及 0.965，屬於高度一致性。

對於目標種類及目標情緒的 Kappa 值計算會比單純的實體計算需要更清楚的定義，由

於兩者標記是綁定關係的，對於不同標記人員標記的關係需要一套直觀可以對照的方式。對此，我們依照標記的邏輯分別對目標種類及目標情緒做出對應的定義。目標種類可以由其標記的產品名稱及廠牌名稱作爲關鍵詞組，若兩標記人員在某關係上標記的產品名稱及廠牌名稱相同，則可視爲兩者是在敘述同一件商品，進而可以對應到關係的標記結果；而目標情緒的對應則是若是兩個標記人員標記關係的結果，在產品名稱、廠牌名稱、目標種類相同的情況下，則可以進行目標種類的一致性比較。經上述方法匹配後，目標種類及目標情緒計算出的 Kappa 值分別爲 0.691 及 0.724，皆高於 0.6，落在可接受範圍。

4 家電評論意見分析

4.1 資料處理

根據相關研究可知，在諸多不同的任務中，TABSA 是與我們標記的資料形式最爲相近的，TABSA 也可再細分爲 ACE、ACSC 兩個子任務。因爲標記有產品名稱、產品廠牌、目標種類、目標情緒四項，我們可以很直觀地將四個標記項目分成三個子任務，產品名稱、廠牌名稱對應 NER，目標種類相當於 ACE 任務，目標情緒可用於 ACSC。在進行不同資料的處理前，先將標記的 2,000 個段落隨機切成等分的訓練及測試資料各 1,000 個段落，後續對於不同的任務中的訓練及測試資料，均是從此部分切出的段落生成的。

在 NER 的任務中，我們使用 BIEOS 做爲詞語標籤。針對兩標記組別，由於廠牌名稱及產品名稱的一致性非常高，因此我們將兩組在廠牌名稱及產品名稱標記結果取聯集，避免有漏標的情況發生。而針對 ACE 任務，必須先定義目標詞是什麼。由於根據我們資料標記的形式，直接套用的目標詞會是產品名稱及廠牌名稱的組合。但此時會面臨到的問題是，不一定所有的關係標記都同時標有產品名稱及廠牌名稱，因此我們要將問題目標詞簡化。在觀察後發現，大部分的評論都是針對特定產品，進行單個或多個廠牌的比對，因此我們可以將目標詞縮減成廠牌名稱。定義目標詞後，便要進行兩標記資料合併，我們將兩標記人員標記同樣的廠牌名稱及目標種類進行合併。在 ACSC 任務的部分則是將兩標記人員標記相同的廠牌名稱、目標種類及目標情緒進行合併。最後三項任務合併後的資料數量如表 2，其中 NER 數量單位以段落數爲單位。

4.2 命名實體辨識與情緒分析

在 NER、ACE、ACSC 這三個任務上。我們均採用 BERT 爲基底的模。NER 使用 BERT-BiLSTM-CRF 的模型架構；ACE 及 ACSC 均使用 BERT 的分類模型。對於 ACE 及 ACSC，我們參考 Su(Sun et al., 2019b) 的做法，應用輔助句子分別訓練三個模型。若按照原論文的方法將 ACE、ACSC 照 TABSA 格式產生輔助句子資料共同訓練，會產生過多的無意義輔助句子，因此在實驗中我們將兩者分別訓練。ACE 任務的輔助句子給定資訊僅有廠牌名稱，而 ACSC 任務的輔助句子給定資訊則有廠牌名稱、目標種類。在建立的輔助句子中，我們以僅用廠牌名稱做爲輔助句的輸入作爲基本效能，兩種任務的範例輔助句如表 1。

Table 1: 輔助句子範例

輔助句類型	ACE	ACSC
僅廠牌名稱	三星	三星
NLI-M	三星	三星-品質
QA-M	敘述三星產品的什麼面向	三星產品的品質面向有什麼情感
NLI-B	三星-品質	三星-品質-正向
QA-M	敘述三星產品的品質面向	三星產品的品質面向有正向情感

Table 2: 個別任務資料量

任務	NER	ACE	ACSC
訓練資料	1,000	1,001	1,345
測試資料	1,000	1,045	1,422

在三個任務中，我們均使用 Micro-F1 做爲評估效能好的指標。而對於 ACSC 任務，爲了要考慮在不同目標種類下可能的效能影響，因此在某些實驗中，該任務中也會使用 Macro-F1 作爲指標參考，該 Macro-F1 的定義爲將每個目標情緒中 ACSC 任務的 Micro-F1 效能計算出來後取平均。

在 NER 的實驗結果中，產品名稱及廠牌名稱的的 F1 分別爲 93.6% 及 94.2%，顯示在簡易的 BERT 模型架構上，在識別產品名稱、廠牌已有相當好的成果。ACE 及 ACSC 的任務結果如表 3。首先比較多元分類 (NLI-M、QA-M) 及二元分類 (NLI-B、QA-B) 的結果，可發現無論是在 ACE 或是 ACSC 上，多元分類的任務形式表現都比二元分類好上許多。而比較 NLI-M 及 QA-M 兩種輔助句子的形式，在 ACE 的任務上 NLI-M 高了 0.9%，而在 ACSC 的任務上兩者的表現則是相同。接著與只用廠牌名稱作爲輔助句子的基本效能比

較，在 ACE 的任務上其與 NLI-M 因輸入形式相同因此效能一樣，故不多提。而在 ACSC 任務中，可發現與 NLI-M 或 QA-M 比起來，有 6% 的效能差距，兩者在輔助句子最大的差別就是在於目標種類資訊的有無，因此可知目標種類的資訊是有助於 ACSC 任務的。

Table 3: ACE 及 ACSC 效能 (Micro-F1)

輔助句類型	ACE	ACSC
僅廠牌名稱	**68.1%**	67.6%
NLI-M	**68.1%**	**73.6%**
QA-M	67.2%	**73.6%**
NLI-B	58.4%	69.3%
QA-B	62.0%	70.3%

接著針對 ACSC 的任務進行更深入的分析，分別比較不同目標種類的訓練資料量與效能的關係 (圖 1)，以及每個目標種類的情緒分布與效能的關係 (圖 2)。圖 1，可發現在不同的目標種類下，ACSC 的任務效能差異甚大，從不到 60% 到超過 80% 的結果都有，不同的目標種類的訓練資料有著極大的數量差異，而效能確實會因爲訓練資料少而較低。圖 2呈現出不同目標種類的三種情緒分布都不一樣，但較難直接看出情緒分布與效能之間的關係。

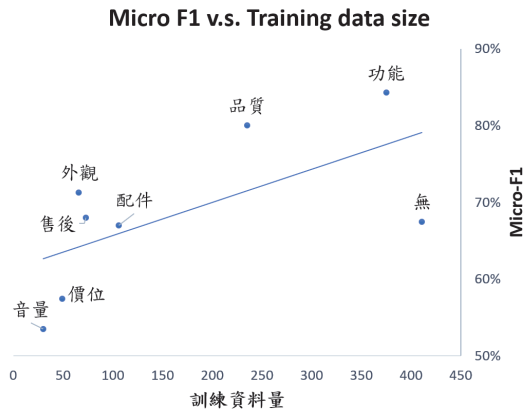

Figure 1: 訓練資料數量與效能關係圖

4.3 多模型方法實驗

由於在不同的目標種類中，情緒的資料存在著不小的差異。因此若能針對各個目標種類都訓練屬於他們自己的 ACSC 模型，說不定能有助於提升效能。所以我們把八個目標種類依照其訓練及測試資料各自訓練 ACSC 模型並評估其效能，而由於在這樣的情況下每個模型都只有包含一個目標種類，所以在輔助句子的建構只需要使用廠牌名稱即可。原先對於所有的種類資練訓練一個模型的方式我們簡稱爲單

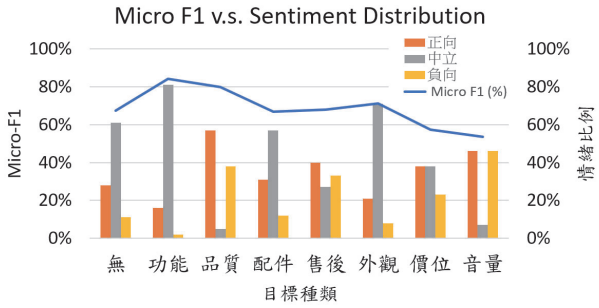

Figure 2: 情緒分布與效能關係圖

模型 (Single Model)，而對於不同目標種類訓練多個模型的方式我們稱爲多模型 (Multiple Models)，各目標種類訓練結果比較如圖 3，可以明顯看出多模型的訓練方式在幾乎所有目標種類上都是輸給單模型的，而多模型的總體效能，Micro-F1 及 Macro-F1 分別 60.1% 與 69.2%，比起單模型的 68.6%、73.6% 低了 8.5% 及 4.4%。在訓練資料特別少的幾個目標種類 (如價位、音量) 成效差距更是超過 10%。由此結果可以推知，多模型的方法在部分目標種類會因爲訓練資料不足而難以得出好的效能，接下來的目標便是要如何提升多模型方法的成效。

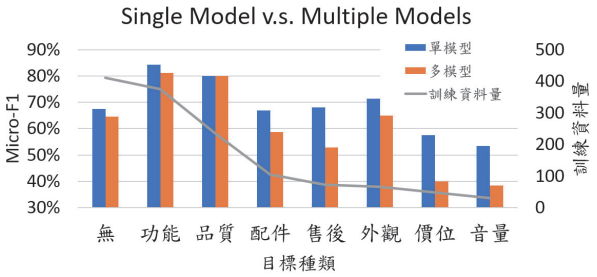

Figure 3: 單模型與多模型效能比較

5 遷移式學習於意見面向分析之應用

在上一個章節中，提到部分的目標種類由於訓練資料過少導致模型效能不好，因此在本章節我們嘗試借助其他的情緒分類資料集，探討其他資料集是否能夠對於我們資料集的訓練有幫助。

5.1 來源資料

遷移式學習除了原始的目標資料集外，也需要來源資料集。由於 ABSA 任務的中文的公開資料集並不多，我們參考 (邱威誠, 2020) 使用的歌手情緒分析資料集，也另外也參考 (Pontiki et al., 2014) 所使用的有關餐廳及筆

Algorithm 1 Reptile

Initialize θ
for iteration = 1,2,... **do**
  Sample tasks τ_1,τ_2,...,τ_n
  for i = 1,2,...,n **do**
    Compute $\theta_i = SGD(L_{\tau_i}, \theta, \alpha, k)$
  end for
  Update $\theta \longleftarrow \theta + \beta \frac{1}{n} \sum_{i=1}^{n} (\theta_i - \theta)$
end for

電的英文評論資料集。前者本質上是 ABSA 任務，但是探討的對象都是歌手；而後者由於是英文評論，因此我們使用 Google 提供的翻譯 API 將其翻譯成中文。而翻譯的資料由於可能產生同樣字詞在不同上下文而有不同翻譯結果的情況，造成文句中的目標詞與實際目標詞不一致的情況，恐會影響訓練，因此在翻譯的過程時我們便把這類的資料剔除，餐廳評論刪除了 47.1% 的資料筆電評論則是刪除 47.6% 的資料。蒐集的統計資料如表 4。

Table 4: 來源資料統計

資料集	正向	負向	中立	資料總數
歌手	28%	50%	22%	8,425
餐廳	59%	17%	24%	1,908
筆電	43%	18%	39%	1,220

5.2 方法

在前個章節遇到的問題中，部分目標種類的訓練資料過少，可視爲是一種小樣本問題，而元學習對於小樣本學習在過去的實驗上有良好的效能。

以元學習中的 Reptile 演算法 (演算法1) 爲例，在最一開始需先初始化元模型的參數 θ，第一、三行分別代表 Outer-loop 及 Inner-loop。在每次進入 Outer-loop 的迭代時，會隨機抽取 n 個任務，再進入 Inner-loop。而 Inner-loop 就會針對每個抽取出的任務計算並以元模型的參數爲初始值更新各自任務模型的參數，其中參數選定更新 k 次後的結果。待每個任務的模型參數更新後結束 Inner-loop，最後更新元模型的參數，更新方向則是取 Inner-loop 中所有訓練過的模型參數與元模型的參數差異平均，更新元模型的參數後再進入下一次的迭代。最終演算法的輸出即是元模型的參數 θ。

我們提出的方法以 Reptile 演算法爲核心，由於來源資料來自不同的領域，甚至不同的語系，撇除不同領域上的資料分布差異，在語言轉換的過程中也可能導致資料分布的改變，因此爲了減少不同來源域的資料差異，我們參考 (Ganin et al., 2016) 搭配對抗式學習的概念，組合成對抗式 Reptile 演算法 (Adversarial Reptile)。由於元學習是基於任務的演算法，在多數使用元學習的環境中會有多個不同的任務。但目前我們要解決的只有 ACSC 任務，因此將元學習演算法中，對於不同任務的定義改變成不同的來源域 (Li et al., 2020)，也就是將每個來源域都視爲是一個任務。在此我們將 BERT 做爲編碼器 (特徵擷取器)，參數爲 θ；分類的輸出層做爲解碼器，參數爲 ϕ；而此外照 (Ganin et al., 2016)，添加了領域判別器，參數爲 γ。解碼器及領域分類器的損失函數皆使用 Categorical Cross-Entropy。

Algorithm 2 Adversarial Reptile

Input: $\alpha, \beta, \lambda, k, D = \{D_1, ..., D_n\}$
Output: θ^0, ϕ^0
Initialize θ, ϕ, γ as $\theta^0, \phi^0, \gamma^0$
for iteration = 1,2,... **do**
  for i = 1,2,...,n **do** // each source D_i
    Compute $(\theta_i', \phi_i', \gamma^0) = SGD(D_i, \theta^0, \phi^0, \gamma^0, \alpha, k)$
  end for
  $\theta^0 \longleftarrow \theta^0 + \beta \frac{1}{n} \sum_{i=1}^{n} (\theta_i' - \theta^0)$
  $\phi^0 \longleftarrow \phi^0 + \beta \frac{1}{n} \sum_{i=1}^{n} (\phi_i' - \phi^0)$
end for

Algorithm 3 SGD (Update base learner)

Input: $D_i, \theta_0, \phi_0, \gamma^0, \lambda, \alpha, k$
Output: $\theta_k, \phi_k, \gamma^0$
Sample $\tau_0, ..., \tau_{k-1}$ from D_i
for j = 0,1,...,$k-1$ **do**
  Compute decoder loss $L^{dec}(\tau_j)$
  Compute discriminator loss $L^{dis}(\tau_j)$
  $\theta_{j+1} \longleftarrow \theta_j - \alpha \nabla_\theta (L^{dec}(\theta_j, \phi_j) - \lambda L^{dis}(\theta_j, \gamma^0))$
  $\phi_{j+1} \longleftarrow \phi_j - \alpha \nabla_\phi L^{dec}(\theta_j, \phi_j)$
  $\gamma^0 \longleftarrow \gamma - \alpha \nabla_\gamma L^{dis}(\theta_j, \gamma^0)$
end for

對抗式 Reptile 的演算法如 Algorithm2、3所示，元 (目標任務) 模型的參數以上標 0 表示，如 θ^0；個別任務模型的參數以下標 i 再加上上標撇表示，如 θ_i'。演算法分爲兩部分，第一部分 (演算法2) 爲外層主要演算法的更新，架構大致上與 Reptile 相同，但在元模型更新時，僅針對編碼器及解碼器進行更新，如演算法2的第 6-7 行，β 代表更新的學習率。領域判別器由於對於所有的個別任務模型都是共用的，因此會在內部迴圈 (第 3-5 行) 更新時進行。

第二部分 (演算法3) 是對於個別任務模型的更新，基本是採用 SGD(Stochastic Gradient Descent) 的更新方式。由於是使用對抗式學習，因此在編碼器及領域判別器中間有一層梯度反轉層。編碼器在更新時的目標即是讓其在確保提升解碼器效能時，同時要混淆領域判別

器，使得編碼器的輸出不會因爲輸入的來源域不同而有過多的差異。如演算法3的第 5 至第 7 行 (α 爲更新的學習率)。領域判別器的參數由於不是在該演算法中主要學習的重點，且所有的個別任務模型都共用同一個領域判別器，因此在此時便會直接更新整個領域判別器的參數。

5.3 實驗

此小節的實驗分成兩個部分，首先比較我們提出的對抗式 Reptile 與前一節單一模型的效能，接著進行消融實驗，由於對抗式 Reptile 可以拆解成 Reptile 及對抗式訓練兩部分，因此我們將其與 Reptile 及 DANN 模型效能進行比較。而本章節的所有實驗結果均是五次實驗數據的平均值。

5.3.1 與原始效能比較

此小節將比較對抗式 Reptile 藉由來源域進行訓練後，所得模型參數在家電產品資料集上使用多模型方式進行微調訓練後得出的效能，與上一節使用單模型方式訓練 (Baseline) 的基礎效能進行比較。各目標種類的效能表現如圖 4，藍色爲 Baseline 效能，橘色爲對抗式 Reptile 效能，可發現除功能及外觀外，在其餘的目標種類，F1 的表現都是對抗式 Reptile 表現較爲優異。因此在加上遷移式學習的方法後，在多模型的效能上確實是可以超越未加入遷移式學習的單模型方法。

而由於圖 4有不少項目是兩者效能極度相近的，因此在每個類別效能的統計檢定比較中，我們檢定兩種方法得出效能差距是否在統計上顯著。使用假設檢定的虛無假設爲兩種方法計算出的效能平均值相等，計算出 p 值並判斷是否小於 0.05，若小於則拒絕假設 (以星號粗體表示)，代表效能差距顯著，反之則接受虛無假設，代表效能差距不顯著。統計結果如表 5所示，其中我們爲每個細部的效能加上其正負兩個標準差的結果，表示其在 95% 的信賴區間中的效能範圍。結果顯示，兩者效能在功能、品質、售後、Macro-F1 四項指標 p 值皆小於 0.05，因此拒絕虛無假設，效果差距顯著，值得一提的是，在功能的部分反倒是原始效能顯著高於我們所提出的模型效能，推測可能的原因是由於在功能層面的敘述上比較沒有特別的固定用詞，情緒也多爲中立，因此在同模型具有較多訓練資料的單一模型訓練環境中就較具優勢；而在配件、外觀、價位、音量四項目標種類可能是因爲 baseline 的效能標準差過大而接受虛無假設；無、Micro-F1 則可能因爲兩者平均效能差距過小而導致差異不顯著。

Table 5: 對抗式 Reptile 效能統計檢定

F1(%)	Baseline	對抗式 Reptile	p 值
無	67.5 ± 2.5	68.3 ± 1.3	0.25
功能	84.3 ± 1.1	82.5 ± 1.8	**0.005***
品質	80.0 ± 5.1	83.9 ± 3.1	**0.02***
配件	67.0 ± 8.1	67.2 ± 4.0	0.91
售後	68.0 ± 9.4	75.6 ± 4.4	**0.01***
外觀	71.3 ± 3.8	70.1 ± 3.3	0.33
價位	57.5 ± 11.1	59.7 ± 2.8	0.40
音量	53.5 ± 7.0	55.1 ± 9.0	0.54
Macro	68.6 ± 5.9	70.3 ± 1.0	**0.04***
Micro	73.6 ± 3.8	74.3 ± 1.1	0.22

Table 6: 消融實驗

多模型	Macro-F1	Micro-F1
DANN	60.7%	69.7%
Reptile	70.0%	74.1%
對抗式 Reptile	**70.3%**	**74.3%**

5.3.2 消融實驗

本節比較的對象是針對對抗式 Reptile 做架構拆解，分爲 Reptile 及對抗式學習的部分，而對抗式學習是 DANN 架構 (Ganin et al., 2016) 的訓練，因此會比較單獨使用 Reptile 演算法及 DANN 兩者模型與對抗式 Reptile 的效能差異。效能如表 6所示。可以明顯看出 Reptile 與對抗式 Reptile 的效能是差不多的，對抗式 Reptile 在兩項指標上只領先 Reptile 不到 0.5%。而兩者皆勝過 DANN 不少，在 Macro-F1 上差距約 10%，Micro-F1 約 5%。因此我們可以得知對抗式 Reptile 在效能上的主要貢獻是來自於其 Reptile 的演算架構，而加上對抗式學習的概念後並未在效能指標上有顯著的成長。

6 結論

在本研究中，我們設計了一份家電產品的資料，可用於 NER、ACE、ACSC 等任務。我們在資料標記完成後也進行了一致性的比對，並確認最終用於實驗的資料是具有較高一致性，品質夠高的。在第一部份的實驗中，對於 NER、ACE、ACSC 三個不同的子任務以 BERT 爲基底模型給出了基礎效能，F1 在 NER 任務的效能平均達到 93.9%，ACE 達到 68.1%，而 ACSC 則有 73.6% 左右的效能。接著針對 ACSC 的任務進行更深入的討論，發現在不同的目標種類之中，ACSC 預測的成效高低不定，細究後推測可能是因爲不同的資料大小或情緒分布導致。因此我們嘗試針對不同的目標種類去各自訓練他們的 ACSC 任務模

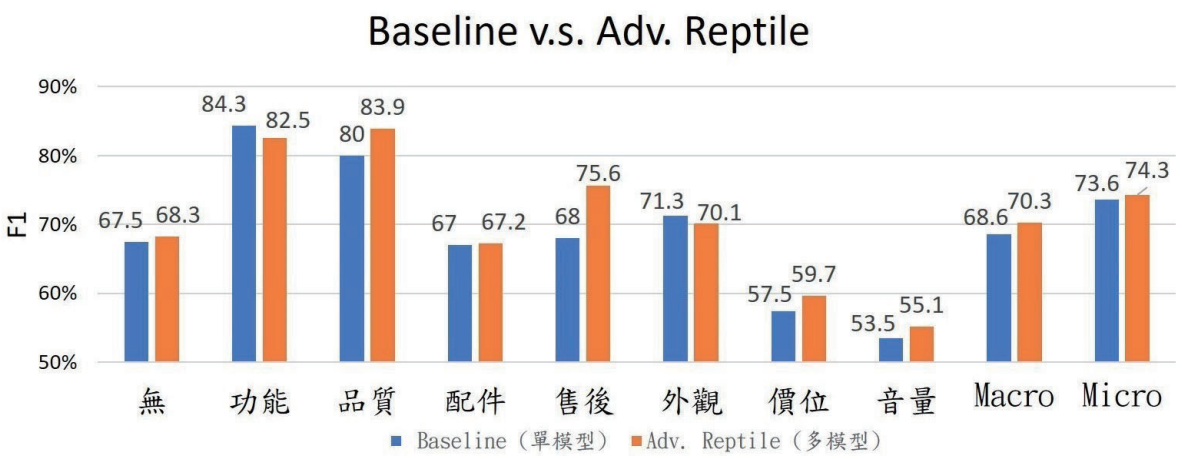

Figure 4: 對抗式 Reptile 與基本效能比較

型，但多模型的方式初始訓練成效遠不如單模型，尤其對於部分目標種類的訓練資料過少，難以訓練出好的效能。接下來的目標是多模型的基礎上提升 ACSC 任務的效能，嘗試使用不同的資料集協助。

而採用不同的資料集協助預測，便會運用到遷移式學習的方法。在此部分的研究，由於元學習對於少樣本資料的訓練相當有效，因此我們在元學習的 Reptile 演算法基礎上，再加上了對抗式訓練的結構，提出了對抗式 Reptile 的模型。在實驗中，對抗式 Reptile 的模型效能在多模型的基礎上可以超越單模型的基準值；在消融實驗中，則觀察到在對抗式 Reptile 的算法中，其架構主要貢獻都是來自於 Reptile，對抗式學習在其中並沒有起到非常大的作用。

元學習在 NLP 相關領域的研究目前資源還比較少，因此本論文藉由元學習探討其在 ACSC 任務上的發展也提供未來 NLP 應用元學習方法的一個方式，期望未來元學習在 NLP 上的發展能更有突破。

References

Dr. S. Sarawathi A. Mounika. 2019. Classification of book reviews based on sentiment analysis: A survey. *IJRAR*, 6.

Harsh Chheda Kiran Gawande Aashutosh Bhatt, Ankit Patel. 2015. Amazon review classification and sentiment analysis. *IJCSIT*, 6:5107–5110.

Manuel Rodriguez-Diaz Ayat Zaki Ahmed. 2020. Significant labels in sentiment analysis of online customer reviews of airlines. *MDPI*.

Alex Brandsen, Suzan Verberne, Milco Wansleeben, and Karsten Lambers. 2020. Creating a dataset for named entity recognition in the archaeology domain. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4573–4577.

R. Caruana. 1998. Multitask learning. In *Encyclopedia of Machine Learning and Data Mining*.

Yaroslav Ganin, E. Ustinova, Hana Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. 2016. Domain-adversarial training of neural networks. *ArXiv*, abs/1505.07818.

Jui-Ting Huang, J. Li, Dong Yu, L. Deng, and Y. Gong. 2013. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7304–7308.

Jing Li, Shuo Shang, and Ling Shao. 2020. Metaner: Named entity recognition with meta-learning. *Proceedings of The Web Conference 2020*.

Mary McHugh. 2012. Interrater reliability: The kappa statistic. *Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB*, 22:276–82.

Quoc Thai Nguyen, Thoai Linh Nguyen, N. Luong, and Quoc Hung Ngo. 2020. Fine-tuning bert for sentiment analysis of vietnamese reviews. *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 302–307.

Alex Nichol and John Schulman. 2017. Reptile: a scalable metalearning algorithm. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

C. Sun, Xipeng Qiu, Yige Xu, and X. Huang. 2019a. How to fine-tune bert for text classification? *ArXiv*, abs/1905.05583.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019b. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. 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)*, pages 380–385. Association for Computational Linguistics.

Mihaela Vela Walter Kasper. 2011. Sentiment analysis for hotel reviews. *Computational Linguistics-Applications Conference*, pages 45–52.

邱威誠. 2020. 應用歌手辨識及情感分析於目標情感偵測與分析之研究. Master's thesis, National Central University, Taiwan.