

結合卷積神經網路與遞迴神經網路於推文極性分類

Combining Convolutional Neural Network and Recurrent Neural Network for Tweet Polarity Classification

葉致廷 Chih-Ting Yeh

國立中山大學資訊工程學系

Department of Computer Science and Information Engineering

National Sun Yat-sen University

M063040008@student.nsysu.edu.tw

陳嘉平 Chia-Ping Chen

國立中山大學資訊工程學系

Department of Computer Science and Information Engineering

National Sun Yat-sen University

cpchen@mail.cse.nsysu.edu.tw

摘要

隨著網路日漸發達，不分男女老少均常常使用社群網站來分享生活瑣事或是評論時事，每天產生的訊息量十分可觀，若我們去分析這些資訊得到社會大眾對事件的好惡，便可以更容易地去做較佳的決策。本文以 Twitter 做為研究對象，針對英語推文進行情緒分析，我們使用 Tweepy 從 Twitter 上收集推文，並用於訓練詞向量 (Word Vector)，然後使用卷積神經網路 (Convolutional Neural Network, CNN) 對訓練好的詞向量進行微調，使其帶有情緒特徵，之後再用於遞迴神經網路 (Recurrent Neural Network, RNN) 進行訓練，得出最後極性分類的結果，我們的系統使用 SemEval-2018 Task1: Affect in Tweets 子任務 V-oc 的資料集進行訓練，且與競賽結果做比較，落在大約第五名的成績。

Abstract

With the development of the Internet, male and female, old and young, often use social network to share the trivia of everyday things and comment on current affairs. The amount of information generated every day is very considerable. If we analyze those data to get the impressions from society, we can easier to make better decisions. This paper chooses Twitter as the research subject and conduct sentiment analysis on English tweets. We use Tweepy to

collect tweets on Twitter and use them to train word vector. After that, the trained word vector is fine-tuned to have emotional features by Convolutional Neural Network (CNN) . Then, the fine-tuned vector is used for training in the Recurrent Neural Network (RNN) to get the final polarity classification results. Our system uses the dataset of the subtask V-oc of SemEval-2018 Task1: Affect in Tweets for training. Compared to the results of competition, we are in the fifth place.

關鍵詞：情緒分析，極性分類，詞向量

Keywords: Sentiment Analysis, Polarity Classification, Word Vector.

一、緒論

近年來機器學習蓬勃發展，在文字的極性分類上也有相當的貢獻。在文字方面的任務中，為了讓電腦能理解人類的文字，我們需要將每個詞轉換成詞向量，若兩個詞的詞向量距離愈近則表示兩個詞的意義相近，愈遠則表示意思相差愈遠，Word2vec [1]、GloVe [2]、fastText [3]都是常見訓練詞向量的方法，而訓練出來詞向量的優劣會大大的影響極性分類的結果。在分類模型方面，無論是 CNN 模型 [4]、RNN 模型 [5]，均取得比傳統基於統計的方法更好的成果。在 CNN 模型中，使用不同大小的卷積核（Kernel）代表一次讓神經網路看多少個詞並取出它們的情緒特徵，若卷積核大小為 1 則代表一次看一個詞，若為 2 則一次看兩個詞，並透過最大池化找出對情緒影響最大的詞或詞組，最後得出分類結果。而在 RNN 模型中，則常使用長短期記憶（Long Short-Term Memory，LSTM）[6]來處理，整合整句句子的涵義然後做出判斷。兩種模型架構所注重的重點不同，CNN 偏向找出影響最大的詞或詞組，而 RNN 則是針對整句句子的結果。

在我們的模型系統中，我們結合 CNN 與 RNN 的優點，利用 CNN 偏重於「詞」的特性，在訓練情緒分析的神經網路時允許反向傳播（Back Propagation）至嵌入層的詞向量，使詞向量帶有情緒特徵[7]，來解決詞向量可能正負面情緒詞之間距離過於接近，而使神經網路無法得知該詞為正面情感或負面情感的問題。接著我們使用調整後的詞向量於偏重「句子」的 RNN 模型中訓練。在 RNN 模型中，由於在我們主要做為評估標準的 SemEval-2018 Task1: Affect in Tweets 子任務 V-oc 為七類 +3 到 -3 的情緒分類問題 [8]，+3 表示最正面的情緒、-3 表示最負面的情緒、0 則是中性情緒，而 SemEval-2017 與之前的資料皆為三類情緒分別為 -1、0、+1 的資料，+1 為正面情緒、-1 為負面情緒、

0 為中性情緒，資料分布情形如表一。為了擴展訓練資料，我們先利用 SemEval-2017 [9] 的三類資料訓練三類模型，得到能抓到「句子」的三類情緒特徵 RNN 權重，並將這個權重做為接下來訓練的七類模型的初始權重，使七類模型能先抓出三類的特徵再進而分成七類，以提升準確率，最終我們以七類模型的分類結果為最後的分類結果[10]。

本文主要分為四個部分：第一部分為緒論；第二部分為研究方法，介紹資料的使用方式與實驗流程；第三部分為實驗結果，介紹實驗的設定與結果；第四部分為結論。

表一、使用資料分布情形

名稱	正向			中性	負面		
	1			0	-1		
	3	2	1	0	-1	-2	-3
SemEval-2017-ALL	9434			16279	7203		
SemEval-2018 訓練集	125	92	167	341	78	249	129
SemEval-2018 驗證集	53	35	58	105	34	95	69
SemEval-2018 測試集	137	91	107	262	80	167	93

二、研究方法

(一)、預訓練詞向量

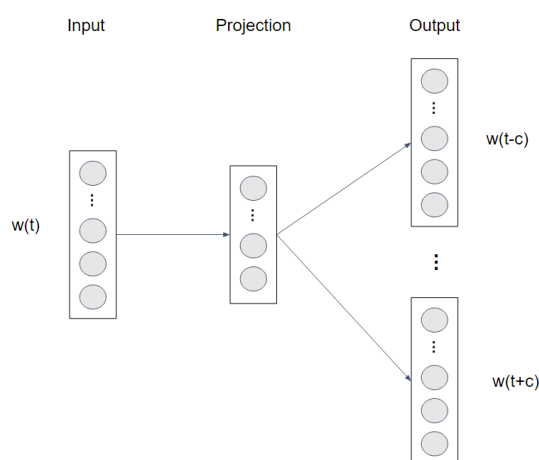
Twitter 上的推文不僅有字數的限制，推文通常都是短短的句子，句子內容包含各種 emoji 與表情符號，文法也不如其他語料庫拘謹，甚至常會有各式英文縮寫出現，如：OMG (Oh My God) 等。因此，若是使用一般語料庫訓練詞向量會有大量的表情符號或網路上流行的縮寫無法包含其中，而這些部分往往都是情緒最強烈的地方，所以我們採用自行從 Twitter 收集推文的方式，藉由 Twitter 官方提供的 Tweepy 這項工具來取得推文，收集期間為 2/27 至 4/13，推文數約 22,000,000 則，詞彙量約 192,000 個詞，表二為英文維基百科語料庫與自行收集之推文資料集在 SemEval-2018 訓練集上是否找到對應詞彙的狀況，可以發現到自行收集的推文資料集在 Twitter 這個文法不拘謹且有非常多新創詞的領域上，表現優於詞彙量為十倍的英文維基語料庫，有更好的對應關係。

表二、英文維基百科與自行收集推文詞彙對應狀況之比較

名稱	詞彙量	未對應詞彙量
英文維基百科語料庫	2,287,131	322 / 4419
Tweepy 自行收集推文	192,307	163 / 4419

同時，我們也對收集的推文前處理[5]，包含全部轉為小寫、標記全大寫強調的詞或 hash tag 的詞、將使用者名稱、網址統一標記為<USER>與<URL>，以減低詞彙的複雜度，之後在訓練情緒分析模型的資料也經一致的前處理，以達到最好的效果。

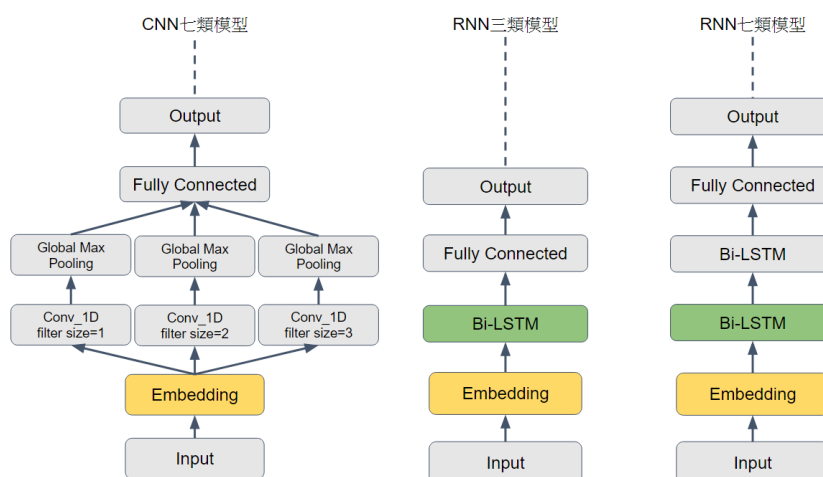
在訓練詞向量的方式上，我們使用 Word2vec 中的 Skip-gram [11]模型來進行詞向量的訓練，如圖一，假設有個句子：You look good in that dress，Skip-gram 會選句子中的一個詞的詞向量為目標詞做為輸入，輸出為整個詞彙表的 softmax 機率，目標詞左右窗口內的詞機率愈大愈好。若目標詞為 good，窗口大小為 $c = 2$ ，則希望輸出 You、look、in、that 的機率愈高愈好，並由此來更新詞向量。但這類訓練詞向量的方式常因正負面情緒詞出現在句子中的位置相近，容易將正反義詞誤認為相似的詞而使兩者之間詞向量的距離十分接近，如有另一個句子：You look bad in that dress，這也是一個合理也常見但意義卻完全相反的句子，在選擇 bad 做為目標詞時，左右窗口的詞一樣，也皆希望輸出 You、look、in、that 的機率愈高愈好，使 good 與 bad 的詞向量會十分的接近，這點不利於情緒分析的任務，在下一小節中，我們將會減緩這個問題。



圖一、Skip-gram 模型架構

(二)、七類推文極性分類系統

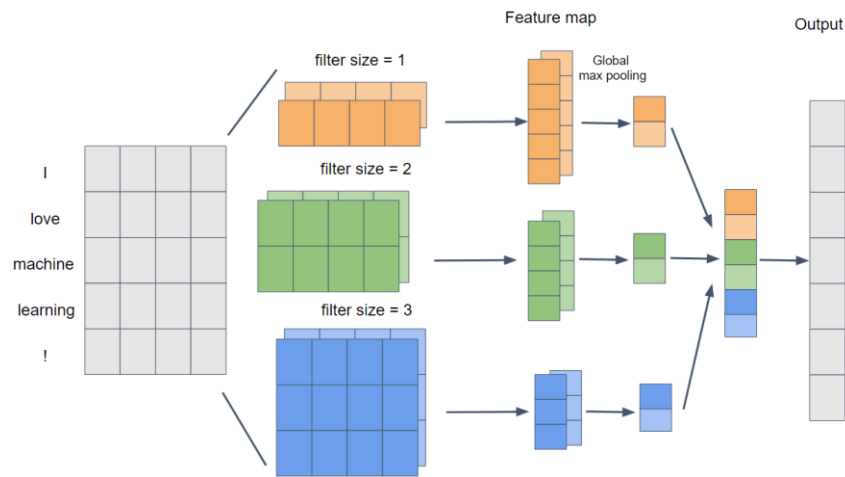
我們分類七類情緒共使用三個模型，分別是 CNN 七類模型、RNN 三類模型以及 RNN 七類模型，整體模型架構圖如圖二。



圖二、推文極性分類整體架構圖

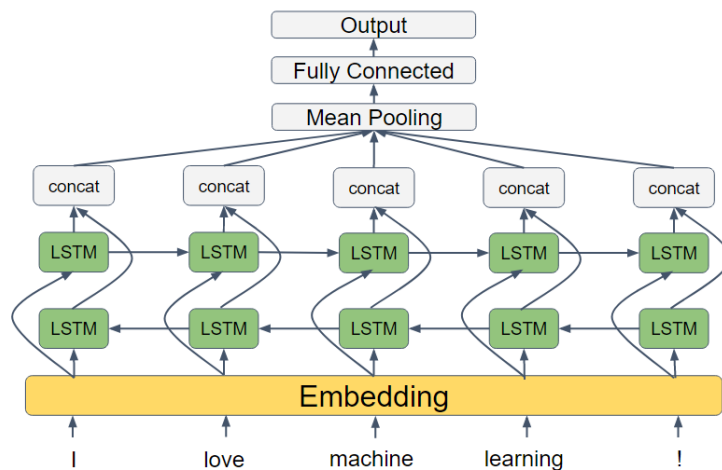
CNN 七類模型如圖三所示，在這個模型中，我們使用 SemEval-2018 Task1: Affect in Tweets 子任務 V-oc 之七類訓練資料，但這個模型並非得到最終分類結果的模型，而是藉由此模型來對嵌入層（Embedding Layer）中的詞向量進行微調，用反向傳播至嵌入層的方式，使正面情緒與負面情緒的詞向量不再那麼接近，並將這份調整過的詞向量用於接下來 RNN 模型的嵌入層上，如圖二中黃色部分所示，以幫助我們在之後的訓練過程中，更好的進行情緒的分類。

在此模型，在詞嵌入後經過大小分別為 1、2、3 的卷積核，相當於以一元語法 (Unigram)、二元語法 (Bigram)、三元語法 (Trigram) 的角度來看推文，並各自取 Global max pooling 找出影響最大的 n-gram 並串接在一起，最後得出分類結果，且允許反向傳播至嵌入層，調整詞向量，並將這份調整過後的詞向量給後續的模型使用。



圖三、CNN 七類模型

RNN 三類模型如圖四所示，所使用的資料為 SemEval-2017 的三類資料，此模型由一層的雙向長短期記憶 (Bidirectional LSTM, Bi-LSTM) 構成，主要的目的為取出 Bi-LSTM 的權重，並用於最後 RNN 七類模型中初始化第一層 Bi-LSTM 的權重，如圖二綠色部分所示，使七類模型能先將推文句子中找出三類的特徵，再藉由第二層的 Bi-LSTM 進一步分成七類的情緒。



圖四、RNN 三類模型

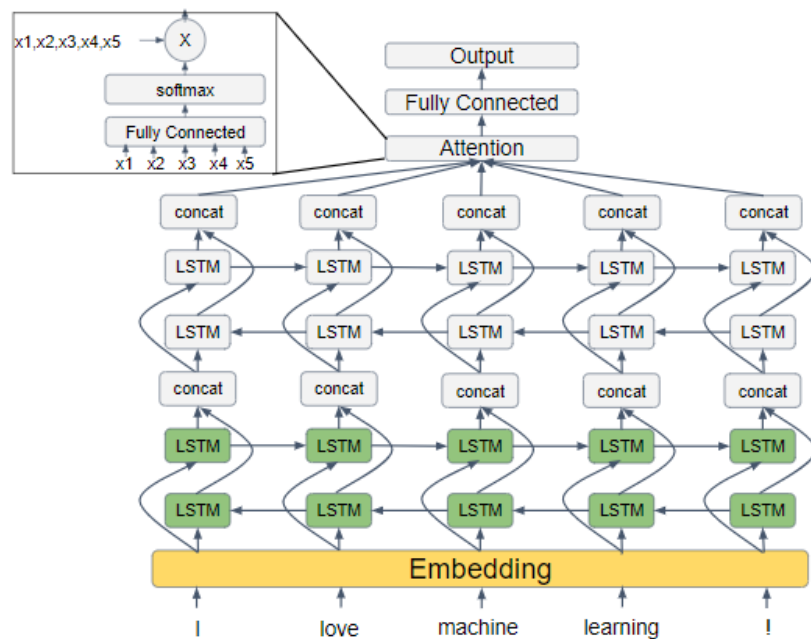
RNN 七類模型如圖五所示，使用資料為 SemEval-2018 的七類資料，也是得出最後極性分類結果的模型，由兩層的 Bi-LSTM 所組成，第一層 Bi-LSTM 由 RNN 三類模型訓練得出的權重初始化，且採用注意力機制 (Attention) [12]，使句子中重要的地方有著較高影響最後極性分類的能力，公式如下：

$$M = \tanh(W_{att} X + b_{att}) \quad (1)$$

$$\alpha = \text{softmax}(\omega^T M) \quad (2)$$

$$r = X \alpha^T \quad (3)$$

假設輸入長度為 N ， d 為第二層 Bi-LSTM 輸出的維度、 $X \in \mathbb{R}^{d \times N}$ 為第二層 Bi-LSTM 各時間的輸出 $[x_1, \dots, x_N]$ 所組成的矩陣、 $M \in \mathbb{R}^{d \times N}$ 、 $W_{att} \in \mathbb{R}^{d \times d}$ 、 $\omega \in \mathbb{R}^d$ 、 $\alpha \in \mathbb{R}^N$ 、 $r \in \mathbb{R}^d$ 。 X 先經過一層的全連接層轉換，且為了保留負值選擇 \tanh 為激活函數，接著再乘以 ω^T 後做 softmax ，得出句子中各位置的權重，句子中較重要能分辨情緒的地方會有較高的權重，最後將 X 乘以注意力權重，即可得到句子經注意力機制整合後的輸出 r 。



圖五、RNN 七類模型

三、實驗結果

本文實驗所使用資料如表一，可以發現 SemEval-2018 的訓練資料相較之下十分稀少，這也是我們需要 RNN 三類模型訓練出來的權重先幫忙做初步分成三類的主要原因，而自行收集的推文數量雖多，但未標記情緒，所以僅用來訓練詞向量。

表三、實驗資料使用狀況

訓練資料	推文數	使用時機
自行收集之推文	22,000,000	預訓練詞向量
SemEval-2017	32,916	RNN 三類模型
SemEval-2018	1,181	CNN 七類模型、RNN 七類模型

在 CNN 七類模型中，共有三個不同大小的卷積核，大小分別為 1、2、3，卷積核的數目均為 200。RNN 三類模型與 RNN 七類模型使用的 Bi-LSTM 正向與反向的維度均為 200 維，雙向合計共 400 維。

在訓練過程中，學習率設為 0.001，且採用 Early stopping 的機制，若連續 10 個 epochs 驗證集的損失沒有下降，則判斷為已收斂並停止訓練，防止過度擬合於訓練資料。三個模型所使用的損失函數皆為交叉熵（Cross Entropy）。

最後所得到的實驗結果如表二，在 SemEval-2018 競賽中使用皮爾森相關係數（Pearson Correlation Coefficient, PCC.）來進行評估，公式如下：

$$\rho_{y_{true}, y_{pred}} = \frac{cov(y_{true}, y_{pred})}{\sigma_{y_{true}} \sigma_{y_{pred}}} \quad (4)$$

分子部分為推文的真實類別與神經網路預測的類別之共變異數（Covariance），標示兩者的相關性，當真實類別為 +3 最強烈正面情緒時，模型預測的類別愈正向則相關性愈高，同樣當真實類別為 -3 時，也同樣模型預測類別愈負面則相關性愈高，分母部分為真實類別的標準差（Standard Deviation）與預測類別的標準差相乘，將共變異數正規化至 +1 到 -1 之間。由實驗結果可以發現，在我們三模型的系統中若少掉任一模型 PCC. 均有下降，若少掉 RNN 三類模型，則七類模型第一層 Bi-LSTM 採用隨機初始化，需從頭訓練無法先抓出三類的特徵；若少掉 CNN 七類模型，則詞向量未經過微調，情緒極性分類的效果下降。在我們的三模型架構下，最終得到 0.800 的 PCC.，對比 SemEval-2018 Task1: Affect in Tweets 子任務 V-oc 的競賽結果，大約落在第五名的成績。

表四、實驗結果與比較

實驗方法	Acc.	PCC.
RNN 七類模型	0.350	0.743
CNN 七類模型 + RNN 七類模型	0.367	0.759
RNN 三類模型 + RNN 七類模型	0.371	0.783
CNN 七類模型 + RNN 三類模型 + RNN 七類模型	0.414	0.800

四、結論

我們使用 CNN 模型來微調詞向量，並使用 RNN 模型先擷取三類句子的情緒特徵，之後藉由七類模型得到最後推文極性分類的結果，結合了 CNN 與 RNN 的特性，PCC.也來到了 0.800。但我們雖然用了 CNN 來舒緩詞向量正負情緒詞可能會過於相似的問題，不過這個問題仍未完全的解決，除了對訓練完的詞向量微調的方法之外，也許可以直接從詞向量的訓練著手，更能根本的解決問題，這也是未來值得研究的課題。

在極性分類的模型方面，文字這類連續的資料非常適合強化學習（Reinforcement learning），且強化學習在其他領域如圍棋的 AlphaGo 等，皆有十分出色的表現，若我們能藉由強化學習找出句子或文字的特徵，相信準確率也會有所提升。

參考文獻

- [1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” arXiv preprint arXiv:1301.3781, 2013.
- [2] J. Pennington, R. Socher, and C. Manning. “Glove: Global vectors for word representation,” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014.
- [3] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” arXiv preprint arXiv:1607.01759, 2016
- [4] H. Hamden, “Senti17 at SemEval-2017 Task 4: Ten Convolutional Neural Network

- Voters for Tweet Polarity Classification,” arXiv preprint arXiv:1705.02023, 2017.
- [5] C. Baziotis, N. Pelekis, and C. Doulkeridis. “Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis,” In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 747-754, 2017.
- [6] F.A. Gers, J. Schmidhuber, and F. Cummins,” Learning to forget: Continual prediction with LSTM”, 1999.
- [7] M. Cliché, “BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs,” arXiv preprint arXiv:1704.06125, 2017.
- [8] S. Mohammad, F. Bravo-Marquez, and M. Salameh, “Semeval-2018 task 1: Affect in tweets,” In: Proceedings of The 12th International Workshop on Semantic Evaluation, pp. 1-17, 2018.
- [9] S. Rosenthal, N. Farra, and P. Nakov. ”SemEval-2017 task 4: Sentiment analysis in Twitter,” In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 502-518, 2017.
- [10] Z.Y. Gao, and C.P. Chen, “deepSA2018 at SemEval-2018 Task 1: Multi-task Learning of Different Label for Affect in Tweets,” In: Proceedings of The 12th International Workshop on Semantic Evaluation, pp. 226-230, 2018.
- [11] T. Mikolov, I. Sutskever, K. Chen, and G.S. Corrado, “Distributed representations of words and phrases and their compositionality,” In: Advances in neural information processing systems, pp. 3111-3119, 2013.
- [12] Y. Wang, M. Huang, and L. Zhao,” Attention-based lstm for aspect-level sentiment classification,” In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 606-615, 2016.