The 2019 Conference on Computational Linguistics and Speech Processing
ROCLING 2019, pp. 25-34


# 結合類神經網路及文件概念圖之文件檢索研究

# Document Retrieval based on Neural Network and Document Concept Graph

盧家馨　Chia-Hsin Lu
國立臺北科技大學資訊工程系
Department of Computer Science and Information Engineering
National Taipei University of Technology
t106598005@ntut.edu.tw

王正豪　Jenq-Haur Wang
國立臺北科技大學資訊工程系
Department of Computer Science and Information Engineering
National Taipei University of Technology
jhwang@csie.ntut.edu.tw

## 摘要

倘若搜尋結果能考慮主題或情境相近的內容，便能搜尋到更符合使用者期待的結果。因此，本研究使用類神經網路及文件概念圖，以探討主題或語意相近之檢索內容，實驗結果顯示，在經由類神經網路所訓練的分類器中，最佳 macro-F1 為 70.0%，而結合文件概念圖的計算後，以查詢內容與結果的相似度而言 nDCG 分數可達 0.959，由此可驗證，基於類神經網路及文件概念圖的結果可以補充和加強資訊檢索的表現。

## Abstract

If the search results can consider topics or similar situations, we can find results that are more in line with user's expectations. Therefore, our research uses neural network and document concept graph to explore the topics or semantics similarity. The experimental results show that the best macro-F1 is 70.0% in the classifier trained via the neural network. Combined with the calculation of the concept graph of the document, the nDCG score can reach 0.959 in terms of the similarity between the search content and the results. This proves that the results based on the neural network and the document concept graph can be used to complement and enhance the performance of information retrieval.

關鍵詞：資訊檢索、類神經網路、文件概念圖、語意相似度

Keywords: Information retrieval, Neural network, Document concept graph, Semantic similarity

## 一、緒論

隨著資訊時代的到來，如何使檢索系統更符合使用者期待，是我們主要的研究主題。為了使檢索系統能搜尋出含有語意或主題相近的結果，我們探討如何取得文件的語意或者主題，本研究旨在增加檢索結果所考慮的因素，例如類別、概念、排序等，期望能提升檢索結果。

我們針對類神經網路預測未知資料之能力進行研究，並利用註釋工具取得概念(Concepts)，建立文件概念圖並探討圖形之間關係，最後我們結合類神經網路及文件概念圖計算，對不同搜尋內容使用不同算法之檢索結果進行討論。透過本研究方法，能經由類神經網路預測文件類別，並藉由文件概念圖的圖形結構關係，找尋具有類別資訊及概念相近之檢索結果，經過實驗驗證，本系統在檢索結果之排名上具有良好的效果，平均 nDCG 分數高於 0.9，此兩種特徵結合確實能輔助我們得到更符合使用者查詢內容的結果。

## 二、相關研究

### （一）. 資訊檢索研究

了解使用者的需求並不容易，如何查詢到適合的結果，是資訊檢索(Information Retrieval, IR)領域持續在研究的其中一個方向。近年來資訊檢索領域逐漸朝著使檢索結果更符合使用者期待的方向發展，本研究嘗試探討檢索結果的相似度，討論何種計算方式能得到與查詢內容更相似的結果。

### （二）. 文件分類研究

對於一般的監督式機器學習文件分類而言，訓練集的資料必須包含所有的類別，然而，此種假設在很多應用並不成立，我們無法確保資料都是系統曾經碰過的類型。此種問題被稱作 open world classification 或 open classification[1]，譯為開放式分類問題。

在開放式分類問題中，近年來已經存在一些可以辨別 unseen 類別的研究，例如[2-4]，Shu Lei[5]等人於 2017 年提出一種名為 DOC 的深度學習算法，經實驗發現，簡單的 CNN 模型在此種類型開放式分類問題上具有良好的效果。Hu Xu 等人[6]於 2019 年，參考了 DOC 架構，在電子商務產品分類進行實驗，解決了辨別新產品類別的問題。由上述研究來看，開放式問題已成為近年來探討議題之一。因此，本研究參考 DOC 架構，探討符合我們研究主題之類神經網路模型，詳細方法將在後續進行說明。

（三）. 文件語意研究

近年來實體、概念搜索已成為 Web 研究的一項重要任務，陸續有研究使用此種方式來探討文件語意對於資訊檢索領域之發展。Yuan Ni 等人[7] 於 2016 年提出了利用概念圖之文件表示，測量文件之間的語意相關性。文件使用多個概念節點(node)來表示，其節點為透過工具從文件中提取的概念。節點之間的邊(edge)代表概念之間的語意和結構關係。此概念圖使用 closeness centrality 對概念進行加權，該權重反映了它們與文件的相關性。Zhenghao Liu 等人[8]於 2018 年提出了一種 Entity-Duet Neural Ranking Model（EDRM），它將知識圖(Knowledge graph)引入神經搜索系統，通過其單詞和實體註釋表示查詢和文件，發現此種知識圖語意顯著提高了神經排序模型的泛化能力。而本研究參考上述研究方法，提出一個基於文件之間關係的文件概念圖，並利用此圖形關係計算文件相似程度。不同於過去方法，我們擷取出文件的多個概念來代表一個文件，並用一個節點來表示，而節點之間的邊代表兩節點之間有共同的概念。

## 三、研究方法

此章節說明研究的方法及架構，如圖一所示，後面章節將針對架構各模組進行說明。

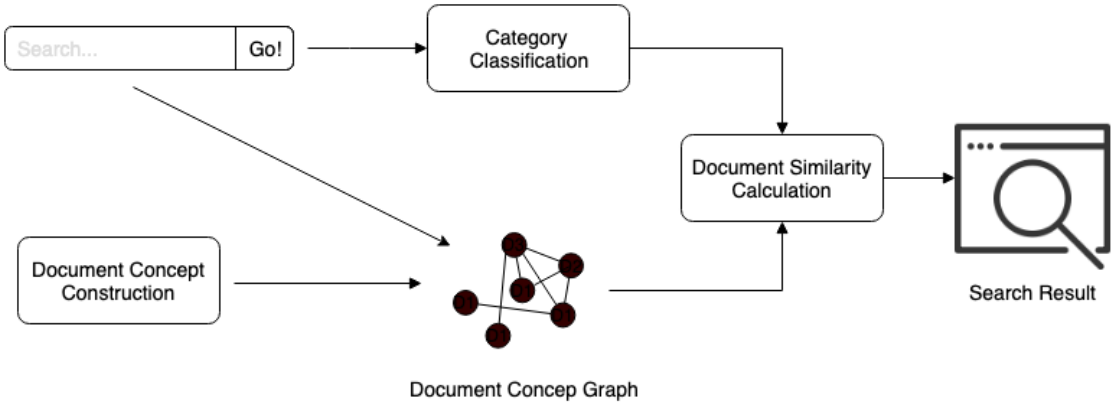


圖 一 、系統架構圖

（一）. Category Classification

本研究參考 DOC[5]利用卷積神經網路(Convolutional neural network, CNN)，此種類神經網路已被實驗證實，對於處理開放式情境問題也具有一定程度的能力，因此，本研究加入此模組，以協助我們探討文件相似度。

1. The Architecture of our CNN model

第一層 Embedding layer 將資料集 D 中的單詞直接 embedding 到密集向量中。近年來在自然語言處理領域上常用的方法會使用 Word2Vec[9]事先訓練一個字詞模型。字詞模型會仰賴不同資料集的特性產生不同向量，所以本研究希望可以不使用字詞模型轉換的方式，僅使用類神經網路的 Embedding layer 計算也能達到良好的效果。

第二層 Convolutional layer，使用不同大小的 filter 分別對密集向量進行卷積，filter 在神經網路中代表對應的過濾器，因此，此計算方式可以得到經由不同過濾條件計算後的結果。

下一步，我們使用 Max-over-time pooling layer 從 Convolutional layer 的結果中挑選最大值以形成 m 維度特徵向量 f，透過兩個 Fully connected layer 和一個中間$ReLU$ activation layer 將 f 降低維度到 n 維向量 x，最後輸出層是應用於 x 的 One-vs-rest 層，此部分在下一小節會做更詳細的描述。

2. One-vs-Rest Layer of CNN

傳統的多分類器使用$softmax$作為最後的輸出層，如此每一個類別的預測機率已經在訓練的時候進行了正規化，便少了彈性調整的能力。因此，我們$sigmoid$函式作為輸

出層，對應的類別採用所有正例的例子，而其餘剩下的例子皆作為反例。

（二）. Document Concept Construction

過去已有研究使用註釋、關鍵字、概念等方式來代表濃縮後的文件[10]。在[7]的研究方法中，提取文件的概念，將概念作為節點，邊代表兩個概念的不同連結關係，並將此概念圖來代表一個文件，計算概念圖的相似來代表文件相似度。而在本研究中，我們參考前述研究方式，將文件參考知識庫取得文件的多個概念，並用多個概念來代表一個文件節點，我們將概念相似的文件建立連結關係，建構文件概念圖，進而利用此圖來計算文件之間的相似度。文件表示為節點$d_i$，利用式 1 及式 2 計算權重$weight$並建立$edge$。我們使用無向圖(Undirected Graph)來建構文件概念圖，並利用相鄰陣列(Adjacency Matrix)來儲存權重($weight$)。

$$weight_{d_i d_j} = number\ \ of\ \ concepts\ \ shared\ \ between\ \ d_i, d_j \tag{1}$$

$$edge(d_i, d_j) = weight_{d_i d_j} \tag{2}$$

（三）. Document Similarity Calculation

此模組我們將結合分類器及文件概念圖來進行檢索相似度計算。建立圖形時，我們將中心點的概念加入文件概念圖，因為其權重代表兩節點相同概念個數，並且我們要得到該圖形結構中擁有最多概念資訊之節點，因此我們計算節點的 Degree Centrality。

1. $Top_{class}$計算

在此小節我們詳列$Top_{class}$的計算方法：

(1)、 將查詢內容輸入分類器，並載入預先訓練好的模型，對查詢內容進行類別預測。

(2)、 利用該類別至文件概念圖找到對應的類別中心點。根據前面對中心點的描述，我們依照鄰居節點多寡進行排序，排序後再挑選出前幾個節點，依據擁有的 weight 多寡再次進行排序，如式 3，找出最後經過兩次排序後最前面的節點，作為類別中心點。

(3)、 取得類別中心點後，我們找出其鄰居節點並依據 weight 高低進行排序，得到與目標節點相似度高的前幾篇文件$Result_C$，如式 4。

$$Center_{class} = (Max(\sum Neighbor(d_i)) \cap (Max(\sum weight(d_i))) \tag{3}$$

$$Result_C = Top_{class}(\ weight(d_i)\ |\ d_i \in Neighbor(Center_{class})) \quad (4)$$

2. $Top_{query}$ 計算

此小節我們詳列$Top_{query}$的計算方法：

(1)、 為了取得整段查詢內容的向量表示，我們使用類神經網路訓練 Doc2vec[11]向量模型，使用該模型透過 word embedding 轉為向量，將查詢內容與文件概念圖的文件向量化。

(2)、 兩向量計算 similarity 找出相似度最高的目標節點，如式 5。

(3)、 取得目標節點後，我們找出其鄰居節點並依據 weight 高低進行排序，得到與目標節點相似度高的前幾篇文件$Result_Q$，如式 6。

$$Target(q) = argmax_i\ \ similarity(q, d_i) \quad (5)$$

$$Result_Q = Top_{query}(\ weight(d_i)\ |\ d_i \in Neighbor(Target(q))) \quad (6)$$

3. $Top_{merge}$計算

此小節我們列出$Top_{Merge}$的計算方法。

(1)、 利用$Top_{Class}$計算的類別中心點，及$Top_{Query}$計算的目標節點，找出兩者的所有的權重關係，合併成一個列表，也就是找出所有的鄰居節點之權重。

(2)、 將此具有權重關係之列表由大到小排列。

(3)、 得到權重最高的前幾篇文件，如式 7。

$$Result_M = Top_{merge}(\ weight(d_i)\ |\ d_i \in Result_C \cup Result_Q) \quad (7)$$

## 四、實驗與討論

### （一）. 實驗資料集

本研究使用的資料集為 DOC[11]所使用的 20 Newsgroups。其內容收集了 18846 篇新聞文件，大致均勻分為 20 個類別；以及 Chen 等人[12]所使用的資料集 50-class reviews，其包含 50 種亞馬遜商品的評論，每一種有 1000 則，總共有 50000 則評論。

### （二）. 文件分類實驗

為了模擬某一筆資料的類別並未出現在訓練集的類別，透過訓練好的模型之後，能夠被正確分類，我們將資料集類別分成 seen、unseen，訓練集資料為 seen 的類別，而測

試集則是所有的類別。

表一 為以 seen 類別搭配不同的資料及進行實驗，此結果為每一個模型重複執行五次後取平均值。其計算結果經過 paired t-test 檢定後，計算出來的 p-value 皆小於 0.01。從此表我們觀察到，每個資料集的 75%類別效果最好，到了 100%的效果卻降低，可能的推斷是資料量多寡會影響模型的效能，抑或是模型訓練 overfitting 的狀況。

表 一、$macro\ F_1 - score$ for datasets

| % of seen classes | 25% | 50% | 75% | 100% |
|---|---|---|---|---|
| 20 Newsgroups | 0.55367 | 0.5948 | 0.62388 | 0.61624 |
| 50-class review | 0.59783 | 0.67283 | 0.70014 | 0.64577 |

（三）. 基於文件概念建圖

我們使用於 2010 年提出的 TAGME[10]作為概念檢測工具，選擇此工具的原因是[13]研究中顯示，TAGME 是各種文檔類型性能最佳的註釋系統，我們列出不同實驗資料集所建構之概念圖的相關屬性，如表二 所示。

表 二、文件概念圖不同資料集屬性表

| 資料集名稱 | 資料集屬性 |
|---|---|
| 20 Newsgroups[14] | Node: 18,846<br>Edge: 128,330,600 |
| 50-class reviews[12] | Node: 50,000<br>Edge: 1,343,168,938 |

（四）. 檢索相似度計算實驗

此小節將分類器及概念圖此兩組模組進行整合計算，我們設計不同的計算方式，並對結果進行討論。後續實驗我們使用的各搜尋編號所對應的查詢內容如下：

1. Semantic Documents Relatedness using Concept Graph Representation.
2. Apple newest product launch
3. how about today's weather
4. convolution neural network
5. machine learning

編號 1 為論文的名稱，較偏學術用語；而編號 2 具有時間、公司名稱、內容資訊等描述；編號 3 為一般使用者日常查詢的內容及用語；編號 4 及編號 5 皆使用了近年來人

工智慧計算之相關用詞。

取得幾組不同算法的結果後，我們將查詢內容與第 $i$ 個檢索文件利用 Doc2vec 向量模型，透過 word embedding 轉為向量，並計算 cosine similarity 作為該位置的相關係數 $rel_i$，如式 9，其中$D_s$為查詢內容，$D_i$為第 $i$ 個檢索文件，最後我們計算正規化折扣累計獲益( Normalized Discounted Cumulative Gain, nDCG )分數。

$$rel_i = cosine\ \ similarity(D_s, D_i) \tag{9}$$

表 三、前 20 篇文件並使用 50%分類器之 nDCG 表

| 搜尋編號 | $Result_C$ | $Result_Q$ | $Result_M$ | Average |
|---|---|---|---|---|
| 1 | 0.862 | 0.901 | 0.918 | 0.893 |
| 2 | 0.937 | 0.959 | 0.902 | 0.932 |
| 3 | 0.928 | 0.886 | 0.922 | 0.912 |
| 4 | 0.935 | 0.912 | 0.913 | 0.92 |
| 5 | 0.912 | 0.909 | 0.921 | 0.914 |
| Average | 0.915 | 0.913 | 0.915 | |

表三 顯示前三種算法取前 20 個檢索結果之 nDCG 分數，可以觀察到普遍的 nDCG 分數皆高於 0.9，可見此系統具有良好的檢索效果。

針對$Result_C$而言，編號 2 的效果最好，或許是此類型的文件得到的檢索結果彼此主題較相近，在取得搜尋結果時可以找到概念較相近的文件。而編號 1 的效果較差，推論是因為敘述較接近學術用語，在新聞資料的分類上效果較不好，另一方面，代表該類型的文件得到的搜尋結果彼此主題內容關係較弱。

下一步，我們觀察$Result_Q$，如同前一種算法，編號 2 的效果最佳，可見此類型的查詢內容，不管是分類器或者是文件概念圖的表現都較良好。而編號 3 在此處表現較差，代表該句子中的四個單字並不能很明確的取得其概念，所以得到的檢索效果較差。

第三種$Result_M$，利用$Top_{Class}$的類別中心點及$Top_{Query}$的目標節點，再合併計算，可以理解為以查詢內容直接使用文件概念圖所得到的結果，再藉由分類器所預測的類別輔助，加強檢索效果的可靠度，因此，我們期望$Result_M$的檢索表現大於$Result_Q$，而結果顯示，除了編號 2 之外，$Result_M$在其他編號的分數的確都大於$Result_Q$，而就總體平均值來看$Result_M$ 也比$Result_Q$高了 0.2%，因此，此實驗結果符合我們的期望。

編號 2 在個別$Result_C$及$Result_Q$的效果都比其他查詢內容好。

然而，以五個編號整體平均值來說，編號 2 的效果還是最佳的，可見雖然合併之後效果降低一些，但整體而言其查詢內容仍能檢索到較好的結果，而編號 1 則是效果最差的，代表其語句上的用詞在分類器及文件概念圖上的效果都表現得不是很好。

## 五、結論

本研究提出結合類神經網路模型及文件概念圖的計算得到檢索結果。在類神經網路的部分，經過實驗驗證，證明此分類器具有一定的分類效果，我們利用得到的類別與文件概念圖整合計算，得到第一組結果；而在文件概念圖的部分，本研究考慮知識庫取得文件的概念，並利用概念來代表文件，計算文件之間的概念相似以建立關係圖，計算時我們直接將查詢內容輸入文件概念圖，得到第二組結果；最後我們合併前面兩個模組在文件概念圖計算的目標點，經由計算得到第三組結果。最後討論三組計算方式對於不同查詢內容之效果，平均 nDCG 分數高於 0.9，證實本研究方法確實具有不錯的效果，得到的檢索結果包含類神經網路預測過的類別資訊，並涵蓋知識庫概念相似。

本研究所提出的方法及計算流程尚有許多改善空間，並且仍有一些限制。

我們使用的卷積神經網路仍有其他多種變形，網路層架構及參數調整更深入的探討或許能提升未知內容的分類效果。在深度學習中，還有很多類型的類神經網路可以進行實驗，或許也能增加文件分類的準確度。

本研究採用的概念檢測工具目前只支援三種語言，英語、德語、義大利語，因此，若是想取得其他語言文件的概念，必須抽換概念檢測工具進行實驗比較。倘若可以加入文件其他特徵，或許能改善文件概念圖的效果。

未來可以實驗更具有評估檢索效果能力之資料集，或者採用資訊檢索公開標準 benchmark 的資料，以多方探討此種結合方式之檢索效果。

## 參考文獻

[1] L. Shu, H. Xu, and B. Liu, "Unseen class discovery in open-world classification," in

*6th International Conference on Learning Representations*, 2018.

[2] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1563-1572.

[3] G. Fei and B. Liu, "Breaking the closed world assumption in text classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 506-514.

[4] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *European Conference on Computer Vision*, 2014: Springer, pp. 393-409.

[5] L. Shu, H. Xu, and B. Liu, "Doc: Deep open classification of text documents," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2911-2916.

[6] H. Xu, B. Liu, L. Shu, and P. Yu, "Open-world Learning and Application to Product Classification," in *The World Wide Web Conference*, 2019: ACM, pp. 3413-3419.

[7] Y. Ni *et al.*, "Semantic documents relatedness using concept graph representation," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016: ACM, pp. 635-644.

[8] Z. Liu, C. Xiong, M. Sun, and Z. Liu, "Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, vol. 1, pp. 2395-2405.

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

[10] P. Ferragina and U. Scaiella, "Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, 2010: ACM, pp. 1625-1628.

[11] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188-1196.

[12] Z. Chen and B. Liu, "Mining topics in documents: standing on the shoulders of big data," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014: ACM, pp. 1116-1125.

[13] M. Cornolti, P. Ferragina, and M. Ciaramita, "A framework for benchmarking entity-annotation systems," in *Proceedings of the 22nd international conference on World Wide Web*, 2013: ACM, pp. 249-260.

[14] K. Lang, "Newsweeder: Learning to filter netnews," in *Machine Learning Proceedings 1995*: Elsevier, 1995, pp. 331-339.