電影檢索系統 MOVIE SEARCH SYSTEM

資訊檢索與文字探勘導論期末專題

劉姝豆
資訊管理學系碩士班一年級
國立臺灣大學
臺北市, 臺灣
r12725045@ntu.edu.tw

劉倍嘉
資訊管理學系碩士班一年級
國立臺灣大學
臺北市, 臺灣
r12725057@ntu.edu.tw

賴煒奇
資訊管理學系碩士班一年級
國立臺灣大學
臺北市, 臺灣
r12725052@ntu.edu.tw

侯岳昇
資訊管理學系碩士班一年級
國立臺灣大學
臺北市, 臺灣
r12725047@ntu.edu.tw

李承軒
資訊管理學系三年級
國立臺灣大學
臺北市, 臺灣
b10705045@ntu.edu.tw

***簡介*—本研究旨在探討傳統電影搜尋機制的局限性，使用 Boolean retrieval 方法比對電影 metadata，並引入影評資料以個人化搜尋及排序。研究利用 TF-IDF、Word2Vec 和 Doc2Vec 三種資訊檢索模型，比較它們在電影搜尋精準度上的表現。**

關鍵字—檢索系統, TF-IDF, Word2Vec, Doc2Vec,

I. 介紹

本研究旨在優化電影搜尋機制，挑戰傳統的 Boolean retrieval 方法。傳統機制常僅利用 metadata 比對，無法滿足對電影劇情特色的細緻需求。為此，我們聚焦於結合影評資料和機器學習演算法，提供更個人化且精確的電影搜尋體驗。

過去的研究主要著重在影評的應用，結合 learn to rank、協同過濾等演算法，以實現更智能的電影推薦系統。然而，本研究更進一步探討了 TF-IDF、Word2Vec 和 Doc2Vec 三種資訊檢索模型的應用。這些模型分別基於詞頻、詞嵌入和文件向量化，用以提高搜尋的精確性。

我們選擇使用 IMDB 提供的前 1000 筆評分最高電影資料集，包含導演、演員、電影種類等 metadata，以及電影名稱和大綱。研究假設使用者主要透過電影大綱和名稱進行搜尋。

研究方法包括資料前處理，使用 Python 實現 PorterStemmer、tokenization 和 stopwords removal。隨後，我們分別採用 TF-IDF、Word2Vec 和 Doc2Vec 作為資訊檢索模型，觀察其對搜尋結果的影響。

實驗結果顯示 Doc2Vec 在查詢精確性上表現最佳，而透過調整 Word2Vec 和 Doc2Vec 模型的 hyperparameter，我們進一步提升了模型的性能。總體而言，本研究為電影搜尋系統的優化提供了有益的實證和參考。

II. 文獻回顧

[1]傳統的電影搜尋機制使用 Boolean retrieval 比對每一部電影的 metadata，而當欄位沒有相符資料時便無法得到結果。因此許多電影檢索的研究聚焦在如何運用影評資料，結合 learn to rank、協同過濾等演算法，進行個人化的搜尋或排序。例如當使用者搜尋令人意想不到 surprising)、或者具有逆轉結局 (twist endings) 的電影，一部符合需求的電影未必會在摘要包含「surprising」或「twist endings」。因此，在回顧文獻後，本研究決定使用 TF-IDF、Word2Vec 以及 Doc2Vec 作為資訊檢索模型，去比較哪種模型在電影搜尋上比較精準。

III. 資料來源

本研究的資料來源為 IMDB 上所提供的電影資料集。在這份資料集中，IMDB 提供了前 1000 筆評分最高的電影。在這份資料中，包含導演名字、演員名單、電影種類以及電影大綱等資料。在本研究的假設中，我們認為使用者最常使用的方法應當是以劇情關鍵字搜尋電影。因此，本研究決定以電影大綱以及電影名稱作為我們主要的資料來源。

本研究所使用的資料形式如下：

- movie title（電影名稱）：

  e.g., The Shawshank Redemption

- overview（電影大綱）

  e.g., Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.

IV. 研究方法

在本次的研究中，主要使用 python 作為程式語言。在並在使用 PorterStemmer、tokenization 以及 stopwords removal 去除無關資訊以得到前處理後資料。

接著，在分別將前處理後資料以三種模型：TF-IDF、Word2Vec 以及 Doc2Vec 作為資訊檢索模型，並對其提交 query 後的得到回傳結果。隨後，從三種模型的回傳結果中各自取出 5 個相似度最高的答案觀察。

A. TF-IDF

TF-IDF（Term Frequency-Inverse Document Frequency）是一種用於資訊檢索和文本挖掘的常用權重算法。它用於衡量一個詞（term）在文檔集中的重要性，通常應用在自然語言處理的任務中，如文本相似度計算、檢索系統和推薦系統。

以下是 TF-IDF 的主要概念：

- 詞頻（Term Frequency，TF）：

資料來源： https://developer.imdb.com/non-commercial-datasets/

詞頻表示一個詞在文檔中出現的頻率。它是指在特定文檔中詞 t 出現的次數，一般可以使用下面的公式表示：

$$TF(t,d) = \frac{\text{詞 t 在文檔 d 中的出現次數}}{\text{文檔 } d \text{ 的總詞數}} \quad (1)$$

- 逆文檔頻率（Inverse Document Frequency，IDF）：

逆文檔頻率表示一個詞對於整個文檔集的重要性。它是詞頻的逆，用於降低常見詞對整個文檔集的影響。IDF 的計算方式是：

$$IDF(t,D) = \log\left(\frac{\text{文檔集 D 的總文檔數}}{\text{包含 } t \text{ 的文檔數}+1}\right) + 1 \quad (2)$$

其中，分母加 1 是為了避免分母為零。

- TF-IDF 的計算：

TF-IDF 是詞頻和逆文檔頻率的乘積，用於獲得一個詞在文檔集中的權重。該值越大，表示詞對於特定文檔的重要性越高。

$$TF-IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

在本研究中，TF-IDF 被應用於電影概述的文本數據。程式碼中讀取了預先計算好的 TF-IDF 矩陣、詞典和 IDF 數據，並使用用戶輸入的查詢文本計算相似度，最終返回相似度最高的前 5 部電影。最後，我們藉由此結果來判斷該模型的表現好壞。

B. Word2Vec

[2]Word2Vec（Word to Vector）是一種詞嵌入（Word Embedding）技術，它能夠將單詞映射到一個連續向量空間。這種技術的核心思想是通過訓練神經網絡模型，使得具有相似語義的詞在詞嵌入空間中距離較近。

Word2Vec 背後的基本假設是「上下文相似性假設」：在文本中，相似的上下文中的詞具有相似的含義。Word2Vec 的主要目標是學習一種映射，將每個詞映射為一個高維向量，使得這個向量能夠捕捉詞彙之間的語義相似性。

Word2Vec 有兩種主要的模型結構：

- CBOW（Continuous Bag of Words）：

CBOW 模型嘗試通過給定上下文詞來預測目標詞。換句話說，給定周圍的詞，預測中間的目標詞。

- Skip-gram：

Skip-gram 模型則與 CBOW 相反，它嘗試通過給定目標詞，預測上下文詞。換句話說，給定中間的目標詞，預測周圍的上下文詞。

在訓練過程中，Word2Vec 通過優化詞嵌入向量，使得模型在給定上下文或目標詞時的預測概率最大。這樣，模型就能夠學到詞彙之間的語義相似性，並將相似的詞在詞嵌入空間中表示為相近的向量。

在 Word2Vec 模型訓練時，有一些的 hyperparameter 需要調整：

- vector_size：

詞嵌入向量的維度大小。這影響著詞的表示空間的維度，較大的維度能夠更好地捕捉詞彙間的語義特徵，但需要更多計算資源。

- window：

考慮上下文詞的窗口大小。這決定了模型在訓練過程中考慮多遠距離的上下文詞。較小的窗口關注局部上下文，較大的窗口能夠捕捉更遠的全局上下文。

在本研究中，使用 Gensim 庫中的 Word2Vec 模型進行了詞向量的訓練，考慮了文本中的上下文信息。TF-IDF 資訊用於計算每個文檔的聚合向量。最後，根據用戶的查詢，計算查詢向量與文檔向量的餘弦相似度，並輸出相似度最高的 5 個電影名稱。最後，我們藉由此結果來判斷該模型的表現好壞。

另外，本研究中也通過更改 window 以及 vector_size 以觀察其對模型表現的影響。(Tuning)

C. Doc2Vec

[3]Doc2Vec（或稱為 Paragraph Vector）是一種用於將文檔轉換為固定維度的向量表示的技術。它是 Word2Vec 的擴展，專門設計用於處理整個文檔的向量表示。

在 Doc2Vec 中，每個文檔被表示為一個固定維度的向量，並且這個向量捕捉了文檔的語義信息。這種表示方式使得我們能夠比較文檔之間的相似性，同時也允許我們在向量空間中執行類似 Word2Vec 的操作，例如找到與某個文檔向量最相似的文檔向量。

Doc2Vec 模型通常基於神經網絡，其中每個文檔被看作是一個上下文（Context）的一部分。模型在學習過程中不僅學習單詞的嵌入表示，還學習捕捉整個文檔的上下文信息。這使得模型能夠生成具有文檔級別語義的向量表示。

與 Word2Vec 相似，在 Doc2Vec 模型訓練時，有一些重要的 hyperparameter 需要調整：

- vector_size：

詞嵌入向量的維度大小。這影響著詞的表示空間的維度，較大的維度能夠更好地捕捉詞彙間的語義特徵，但需要更多計算資源。

- window：

考慮上下文詞的窗口大小。這決定了模型在訓練過程中考慮多遠距離的上下文詞。較小的窗口關注局部上下文，較大的窗口能夠捕捉更遠的全局上下文。

在本研究中利用 gensim 中的 Doc2Vec 模型實現文件向量化。透過這種方式，每個文檔都可以轉換成一個固定維度的向量表示，捕捉文檔的語義信息。模型透過學習文檔之間的關聯，生成每個文檔的向量表示。在模型訓練完成後，使用新的文檔向量進行相似性計算。這使得模型能夠識別與查詢文檔最相似的其他文檔。最後，輸出相似度最高的 5 個電影。最後，我們藉由此結果來判斷該模型的表現好壞。

另外，本研究中也通過更改 window 以及 vector_size 以觀察其對模型表現的影響。(Tuning)

V. 結果

A. 各模型平均表現

在分別輸入相同的查詢至 TF-IDF、Word2Vec 以及 Doc2Vec 後，人工判定該查詢是否正確，重複數次便能夠得到該模型的平均查詢 precision，如 TABLE I。

TABLE I. 各模型之平均 PRECISION

	TF-IDF	Word2Vec	Doc2Vec
Precision	0.52	0.45	**0.56**

從 TABLE I 中的結果可以得知，Doc2Vec 在 precision 上有最好的平均表現。

B. Word2Vec Tuning

在 Word2Vec 模型中，我們可以調整兩個 hyperparameter: window 以及 vector_size ，並且對不同組合的 Word2Vec 模型分別輸入相同的查詢，重複數次便能夠得到該組合的平均查詢 precision ，如 TABLE II。

TABLE II. WORD2VEC 在各種 HYPERPARAMETER 時的 PRECISON

	window=10	window=5	window=1
vector_size=50	0.36	0.32	0.24
vector_size=100	0.36	**0.40**	0.28
vector_size=200	0.36	0.36	0.24

從 TABLE II 中的結果可以得知，在 window = 5 且 vector_size = 100 時 Word2Vec 在 precision 上有最好的平均表現。

C. Doc2Vec Tuning

在 Doc2Vec 模型中，我們可以調整兩個 hyperparameter: window 以及 vector_size ，並且對不同組合的 Word2Vec 模型分別輸入相同的查詢，重複數次便能夠得到該組合的平均查詢 precision ，如 TABLE III。

TABLE III. DOC2VEC 在各種 HYPERPARAMETER 時的 PRECISON

	window=10	window=5	window=1
vector_size=50	0.56	0.56	0.56
vector_size=100	**0.68**	0.64	0.64
vector_size=200	0.60	0.60	0.60

從 TABLE II 中的結果可以得知，在 window = 10 且 vector_size=100 時 Doc2Vec 在 precision 上有最好的平均表現。

VI. 參考文獻

[1] K. Kurihara et al. "Doc2Vec-based Approach for Extracting Diverse Evaluation Expressions from Online Review Data," The 23rd International Conference on Information Integration and Web Intelligence. 2021.

[2] T. Mikolov et al. "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781 (2013).

[3] Q. Le, T. Mikolov. "Distributed representations of sentences and documents," International conference on machine learning. PMLR, 2014.