文字探勘期末專案書面報告

Group 18 – 電子郵件行銷 (Sneaked spam)

陳鵬仁	林子昕	黃韻文	陳奕兆	丁啟恩
B09201013	B09702032	B09610020	B09705053	B08702039

摘要

本文旨在研究文本改動對於模型判定是否為垃圾郵件的影響，目標是希望經由調整垃圾郵件的內文後降低模型的 recall。選用 Multinomial Naive Bayes 的機器學習模型，研究過程包含「刪除」、「抽換」、「新增」等方式，得出「新增」常出現在一般郵件但未曾出現在垃圾郵件的字（下稱差集字）可以最有效的影響模型的判定。加入頻率最高的 20 個差集字可以降低 recall 高達 50%，為了實務上不顯突兀，也試著人工透過 stopwords 以及這些差集字組成有意義的句子，兩句話共包含 8 個差集字可以使 recall 下降 20%。

1 研究動機

電子郵件行銷具有快速、環保等多項優點，一直以來都是企業愛用的宣傳行銷手法。加上科技的進步，電子郵件行銷所帶來的效益更容易透過後台數據等資料被量化，此種行銷方式也一度成為主流。然而，氾濫的電子郵件行銷可能導致信箱被廣告信件塞爆，導致重要的訊息被遺漏掉，對於一般民眾的生活造成不便。於是各個電子郵件的服務商開始對於信件做出審查，有了檢測的機制以後，許多宣傳的信件被分類到「垃圾郵件」。這可能對於民眾在使用電子郵件上可以說是一大福音，但是對於那些精心分析客戶群想要達到精準行銷的廣告商來說十分可惜。

2 研究目標

我們希望透過機器學習的技術模擬出市面上的電子郵件分類器，並且著重於「信件內文」，以提高 precision [1] 為目標篩選適合的模型。接著我們會對於不同改動內文的方式做研究，希望可以提出合適的方式來降低模型的 recall，以供廣告商作為未來電子郵件行銷的參考。

3 文獻回顧

[1] 能看出垃圾郵件的判定具有非常多元的面向，包含：

[1] 比起讓垃圾郵件被判斷為一般郵件，將一般郵件誤判為垃圾郵件更為嚴重

- 發送方式：大量發送、沒有標題、亂碼
- 寄件人：匿名、網站聲譽、明確的目標對象
- 信件內容：多種顏色與字體、垃圾郵件常見詞彙、大量連結與圖片
- 附件：檔案容量大、多個附件

[2] 垃圾郵件檢測的一些機制包括郵件標頭、主題和正文中提取特徵，通常依照下列順序：正文過濾、標頭過濾、黑名單過濾、基於規則的過濾。

對於模型的選擇，學術上也有許多供參考的案例。[3] 對各種垃圾郵件分類方法進行研究，並以 Naive Bayes, KNN, SVM 等分類器實作。在該研究中發現 Naive Bayes 分類所得到的 precision, recall, accuracy 皆為最高，其原因是在 sns email 中，平均信件長度只有 16。[4] 提到在文本較小的時候 NB 較為精準，尤其是 multinomial Bayes，[5] 但如果 vocabulary size 較大的時候精準度就會開始下跌，[6] 不過其下跌幅度仍小於 SVM 。[7] 中 Naive Bayes 模型結果的 false positive rate 維持在極低的水平，更是所有分類器中運行時間最短的。[8] 實作了以 Naive Bayes 為模型的分類器，當中也說明使用 Naive Bayes 的好處，包含學習容易、速度快。[9] 提到 Naive Bayes 方便針對各產業做出挑整並且提出「加入正常郵件會出現的字到垃圾郵件中」的想法。

此外在 [10] 中提到垃圾郵件發送商每天都在使用新技術去規避關鍵字或是短語使得分類器失靈，或者是一個正常郵件可能出現許多 spam words 而被分類為垃圾郵件 [11]，因此使用 laugage model 去判斷上下文關係是未來發展的趨勢。然而在 [12] 明確告訴我們 n-gram 過於複雜，在時間以及成本上相當不划算。而 [13] 提及使用垃圾郵件特定詞彙的提取作為辨別垃圾郵件的依據可以大大地提昇準確率，因此我們選擇單純使用 term frequency(TF) 以及 spam words 去做研究。

除了學術研究，市場上也有垃圾郵件檢測平台，例如 [14] Mailmeteor。該網站有內建的垃圾郵件常見字詞，並提供使用者即時互動的檢測功能。郵件發送者能在郵件送出之前將內文複製到平台，平台會標示出屬於垃圾郵件常見字詞的文字與當前文案的信譽分數。使用者能在平台上即時的調整，直到文案獲得好的信譽，並且不含有垃圾郵件常見字詞。這個平台受到多個知名網站（如 Spotify）推薦，可見其市場價值。

4 資料集

4.1 資料來源

我們選用 kaggle 上的公開資料，作為這次專案的資料來源。內容包含 3,051 封電子郵件，其中包括 2,551 封一般郵件以及 500 封垃圾郵件。

4.2 資料型態

資料集包含下列幾種不同型態的電子郵件：

- text/plain: 2,674 筆
- text/html: 1,81 筆
- multipart/signed: 74 筆
- multipart/alternative: 56 筆
- multipart/mixed: 53 筆
- multipart/related: 11 筆
- multipart/report: 2 筆

4.3 資料前處理

資料內容保存所有的 email headers（圖 1），我們經由 python 的 `email`、`email.policy` 以及 `BeautifulSoup` 套件，得到去除 headers 的內容（即信件內文）(圖 2)。因為 `BeautifulSoup` 並不能成功解析 multipart 型態的電子郵件，於是我們只選用部份的電子郵件（text/plain、text/html）作為我們訓練模型的資料。

圖 1: raw data

圖 2: text-only

5 訓練模型

接著我們將進行資料前處理後的資料進行模型訓練。考量到原始資料集中垃圾郵件和一般郵件資料數量的差異，我們僅抽取 400 筆垃圾郵件和 400 筆一般郵件進行模型訓練，以高 precision 為目標用 k-fold validation 進行參數的調整，並以 100 筆垃圾郵件和 100 筆一般郵件進行測試。

5.1 Multinomial Naive Bayes

首先，我們嘗試以 Multinomial Naive Bayes 訓練模型。在經過 validation data 的測試之後，我們將模型的參數調整為 `fit_prior=False`。訓練結果如下表（表 1）。

	precision	recall	f1-score	support
0（一般郵件）	0.96	1.00	0.98	100
1（垃圾郵件）	1.00	0.96	0.98	100
accuracy			0.98	200
macro avg.	0.98	0.98	0.98	200
weighted avg.	0.98	0.98	0.98	200

表 1: Multinomial Naive Bayes

5.2 Bernoulli Naive Bayes

接著，我們以 Bernoulli Naive Bayes 模型進行測試。經過 validation data 測試後，我們將參數調整為 `fit_prior=False`。訓練結果如下表（表 2)。

5.3 TFIDF to Support Vector Machine

接著，我們嘗試以 Support Vector Machine 對郵件進行分類。我們先將

	precision	recall	f1-score	support
0（一般郵件）	0.96	0.25	0.40	100
1（垃圾郵件）	0.57	0.99	0.72	100
accuracy			0.62	200
macro avg.	0.77	0.62	0.56	200
weighted avg.	0.77	0.62	0.56	200

表 2: Bernoulli Naive Bayes

文件轉為 TFIDF vector 後，將其投入 SVM 模型，在經過 validation data 測試後，決定以 `kernal='rbf'` 的模型進行訓練。訓練結果如下表（表 3）。

	precision	recall	f1-score	support
0（一般郵件）	1.00	0.84	0.91	100
1（垃圾郵件）	0.86	1.00	0.93	100
accuracy			0.92	200
macro avg.	0.93	0.92	0.92	200
weighted avg.	0.93	0.92	0.92	200

表 3: tf-idf to SVM

5.4 SVD to Support Vector Machine

最後，我們嘗試以 SVD 將文件向量壓縮為 10 維，並丟入 SVM 模型。經 validation data 測試後決定以 `kernal='rbf'` 的模型，並將參數調整為 `C=2.0`。訓練結果如下（表 4）。

	precision	recall	f1-score	support
0（一般郵件）	0.99	0.93	0.96	100
1（垃圾郵件）	0.93	0.99	0.96	100
accuracy			0.96	200
macro avg.	0.96	0.96	0.96	200
weighted avg.	0.96	0.96	0.96	200

表 4: SVD to SVM

5.5 最終模型挑選

對於模型的挑選，我們參考了許多垃圾分類器相關的文獻，發現 Naive Bayes 分類器不但學習方式容易且速度快，還能針對產業進行調整且較不易被穿透。此外，針對垃圾郵件的分類準確率高達 95% 到 99.95%，因此大部分對於垃圾郵件的分類都使用 Naive Bayes 的模型分類器進行分類。

而針對實務上的操作，垃圾郵件分類器應避免將應屬一般郵件的郵件分類至垃圾郵件，因此，分類器的選擇上應以垃圾郵件的高 precision 為目標進行模型的挑選。依據我們所訓練的四種分類器結果與文獻回顧進行綜合考量，我們最後決定以垃圾郵件 precision 最高的 Multinomial Naive Bayes 的訓練模型作為本報告後續動作的最終模型。

6 實驗設計

在訓練模型的部份，使用到 400 封一般郵件以及 400 封垃圾郵件當作訓練資料，另外使用到一般郵件和垃圾郵件各 100 封作為測試資料。我們將原本作為測試資料的一般郵件（共 100 筆）放到訓練資料集（一般郵件 500 筆、垃圾郵件 400 筆），並重新訓練成為新的模型。在接下來的方法中，我們只會對原本作為測試資料的 100 封垃圾郵件內文做出更動並再試著丟入新的模型。

6.1 方法一

首先，調整垃圾郵件最直觀的作法是取代垃圾郵件常用的字詞。

我們下載由垃圾郵件檢測平台 mailmeteor[2] 所提供的垃圾郵件常見字詞，內含 769 個常見字詞，並對垃圾郵件做斷詞，找出交集的字詞共 131 個，接續使用 word2vector 中的 `.most_similar()` 查找 131 個字詞的同義詞，由於 word2vector 精確度有待商榷，此查找的結果僅作為參考，最後以人工的方式建立出 131 個垃圾郵件常見字詞的同義字抽換字典，並確認大部分抽換的字都不是垃圾郵件郵件常見字詞，然後將 testing data 中的垃圾郵件以字串處理的方式作抽換，調整好之後重新將這些文本放入訓練好的模型作預測，recall 下降 1%。

進一步思考，垃圾郵件除了常用字詞外，尚有大量使用特殊符號的特徵，因此，我們在原先的同義字字典中加入特殊符號，將其轉換為空字串(移除)，並將調整好的文本重新放入字典，結果 recall 上升 1%。

最後，根據文獻資料提及，垃圾郵件經常使用大量的連結，是分類時的一項指標，因此，我們單獨做了將連結移除的調整，recall 維持不變。

總結而言，調整垃圾郵件常見的字為同義字的方式僅使 recall 下降了 1%；加入特殊符號處理後 recall 不減反增，提升了 1%；單獨移除連結 recall 維持不變（表 5）。此結果不盡理想，我們推測是因為這個做法忽略了與正常郵件的相關性。換句話說，我們挑出了垃圾郵件常見的用字做替換，然而這些用字卻可能也常出現在正常郵件中，因此，在調整文本的同時也降低了與正常郵件的相似度。

方式	同義字抽換	同義字抽換 + 特殊符號移除	移除連結
recall	下降 1%	上升 1%	維持不變

表 5: 方法一總結

6.2 方法二

為改善方法一的缺失，這次我們取出僅存在於垃圾郵件的常見用字做調整。方法為：分別對一般郵件及垃圾郵件做斷詞，並將垃圾郵件出現的字頻減去一般郵件出現的字頻，取出前 100 個相對常出現在垃圾郵件的字。首先嘗試將這些字詞全部刪除，接續嘗試在這些字後面加入特殊符號 (misspelling)，使這些字變成新的不曾出現過的字詞，最後也嘗試了以同義字做抽換，三種方式的結果如下 (表 6)。

方式	移除高頻字	故意拼錯高頻字	以同義字抽換高頻字
recall	下降 2%	維持不變	維持不變

表 6: 方法二總結

方法二的做法依舊不盡理想，皆未使 recall 有顯著的下降。此時我們重新回顧相關的文獻資料，發現除了抽換垃圾郵件常見字詞，還有另一種較不直觀的做法是將一般郵件的高頻字放入垃圾郵件中混淆模型的判別。

6.3 方法三

延續方法二的結論，我們對垃圾郵件及一般郵件做斷詞，取出只出現在一般郵件的字並挑選前 20 個高頻字

[2] https://mailmeteor.com/spam-checker

放入待調整的文本，以訓練好的模型重新預測，此時的 recall 顯著下降了 50%，加字的做法成功讓一半的垃圾郵件被模型分類為一般郵件。

考量實務上的意義，單純的加入 20 個字在文本中不符合常理，因此，我們將範圍擴大，找出一般郵件與垃圾郵件差集後的前 55 個高頻字，以人工的方式挑選出 8 個字詞，與 stop-word 混和成兩句有實質意義的話，將兩句話加入文本末端後重新以模型分類，此時 recall 下降了 20%（表 7）。

方式	加入 20 個高頻字	加入兩句話（包含 8 個高頻字）
recall	下降 50%	下降 20%

表 7: 方法三總結

7 結論

這份研究尚存有許多限制，包含資料集數量太少造成可能的偏誤、word2vector 精確度不足、手動抽換同義字等等。然而根據前述的實驗，我們發現加入 $\{$一般郵件$\} \setminus \{$垃圾郵件$\}$ 的高頻用字是最有效率的做法。因此，加字的調整方式將作為此份報告最終得出的解方。未來若有電子郵件行銷的需求，應優先使用加字的方式做調整，接續才考量垃圾郵件常見字詞的同義字抽換，而刪除連結與特殊符號則是相對不具影響性的做法。

為使解方更貼近真實狀況，未來或許還能根據不同產業的廣告信製作出不同的句子，使業主能挑選符合自身主題的句子放入，減少與信件內文的違和程度，此外，也能以反白、放置於網頁下層等方式讓加入的字詞不被人眼看到，卻能被分類模型納入考量。透過這項提案，即能讓廣告業主在不影響原先信件內文的情況下將資訊傳遞給特定的受眾，增加產品或服務的觸擊率。

參考資料

[1] 科技橘報【解密判斷機制】(2021/3/5)
https://buzzorange.com/techorange/2021/03/05/email-ads-esp/

[2] R.Amin, H.Aldabbas, D.Koundal, B.Alouffi, T.Shah (2022) "Machine Learning Techniques for Spam Detection in Email and IoT Platforms: Analysis and Research Challenges"

[3] W.A.Awad, S.M.ELseuofi (2011) "Machine Learning Method From Spam Emails Classification"

[4] P.Navaney, A.Rana, G.Dubey (2018) "SMS Spam Filtering Using Supervised Machine Learning Algorithms"

[5] K.Michael, Schneider (2003) "A Comparison of Event Models for Naive Bayes Anti-Spam E-Mail Filtering"

[6] S.Saab, N.Mitri, M.Awad (2014) "Ham or spam? A comparative study for some content-based classification algorithms for email filtering"

[7] S.K.Trivedi (2016) "A Study of Machine Learning Classifiers for Spam Detection"

[8] M.Abdoh, M.Musa, N.Salman (20 09) "Detecting Spam by Weighting Message Words"

[9] 垃圾郵件終結者－貝氏過濾法 (2004/8/2) https://www.ithome.com.tw/tech/28999

[10] M.Raza, N.Jayasinghe (2021) "A Comprehensive Review on Email Spam Classification using Machine Learning Algorithms"

[11] A.Karim, S.Azam, B.Shanmugam, K.Kannoorpatti, M.Alazab (2019) "A Comprehensive Survey for Intelligent Spam Email Detection"

[12] E.Dada, J.Bassi, H.Chiroma, S.Abdulhamid, A.Adetunmbi, O.Ajibuwa (2019) "Machine learning for email spam filtering: review, approaches and open research problems"

[13] K.Islam, A.Amin, R.Islam, N.Mahbub, I.Showrov (2021) "Spam-Detection with Comparative Analysis and Spamming Words Extractions "

[14] mailmeteor https://mailmeteor.com/spam-checker

[15] Lecture slides.