Project Proposal

0816012 鍾任軒 0816088 陳元挺 0819818 許慈若 0613130 張芷嫣

I. Problem Statement and Task Definition

The information spread on social media often spans across millions of users and makes an impact immediately. Therefore, the authentication of information has been an important issue given that people make decisions based on the information they collect or receive. We want to build a system that detects fake news through machine learning techniques, which aims to help users obtain correct information under the era of information explosion. This fake news detection system is supervised textual-based, it will output whether the news is true or false given the headline and content paragraphs of the news.

II. Description of the challenges

Here we talk about the challenges of fake news detection through several perspectives. First, there are many forms which the news can take. The most common are text-based news with a header and body content. Many news now also include images and videos. Yet, text, images and audio are very different in terms of data structure, and they rely on different methods in the domain of AI to tackle. Therefore, in this project we mainly focus on the essential content of news — "text", where we analyze the authenticity through news header and content.

Second, sometimes there exists a relativity on the judgment of truthiness as it may be based on human's morality, ideology and other perspectives. This higher-level knowledge or connotation poses a challenge for models to acquire.

Last but not least, many fake news related dataset are US-based and are for a certain period. So language can hardly be changed unless other learning techniques are employed. Moreover, certain events are popular over a period, but new events may happen in the future. Thus, the ability to generalize may also be a challenge.

Our task in general can be divided into two parts, analyzing the texts through NLP models where it captures the semantic meaning; then, classifying the processed texts to predict whether it is fake news. Moreover, we have carefully selected a less biased dataset for training in order to address the generalization problem.

III. Input/Output Behavior with Concrete Examples

Input (News header and news content):

Title: Billionaire Donald Trumps Presidential Campaign Is Flat Out Broke

Content: Donald Trump raised just $29 million for his presidential campaign committee in the first 19 days of October, about half as much as his Democratic rival, putting him at a severe financial disadvantage in the crucial final days of the White House contest, new campaign finance reports filed Thursday night showed. ……

Trump has done what he does best. He talked a big game while bankrupting the Republican Party for his own personal gain. Convincing Republicans to give him their nomination may go down in history as Trumps biggest con of all.

Output (Binary: 0/1): 1

Datasets Reference: WelFake Dataset https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9395133

IV. Related Works

以下會從 4 個方面介紹假新聞偵測的方法 [Zhou & Zafarani 2020]，最後我們會聚焦在一種方法上去做研究。

A. Knowledge-based methods: 由於假新聞散播不實的資訊誤導大眾，因此最直接的檢測方法就是將新聞內容的陳述比對已知的事實來檢查其真實性 [Shu et.al 2017]，這個過程稱為事實查核 (Fact checking)，需要依靠大量專業知識以及可信度高的資料來源 (例如：維基百科) 去判別新聞的真假。我們可以透過訓練這些資料來建立監督式機器學習模型，來自動檢測新聞文章 [Zhang & Ghorbani 2020]。

B. Style-based methods: 將文章風格作為判斷假新聞的基準。文章風格可以從分析具體的文本特徵，例如：詞頻 (Bag-of-words/TF-IDF)、詞性 (Part of speech tagging)、修辭 (Rhetorical Structure Theory, RST)、文法 (Probabilistic Context-Free Grammars, PCFG) [Zhang & Ghorbani 2020; Zhou & Zafarani 2020; Shu et.al 2017]，到分析文章的複雜性，像是可讀性、詞彙多樣性 (Type-Token Ratio, TTR) 等。這些特徵反應了作者的寫作特性以及個人態度，藉此可以評估新聞內容的真實程度 [Horne & Adali 2017] 。例如 Wynne & Wint 研究 word n-grams 和 character n-grams 作為檢測假新聞的特徵，實驗結果發現 character n-grams 比 word n-grams 效果更好[Wynne & Wint 2019]。Hu et al. 比較 RST、LIWC、CNN 模型的準確度，並提出 MCNN-TFW 模型，TFW 是指高頻率單字 (sensitive words) 的權重，可以顯示出單字對真假新聞的重要性，透過計算 TFW 結合 MCNN 提取的語意資訊來進行假新聞偵測 [Hu et al. 2020] 。 Verma et al. 設計了 WELFake 模型，提取各種語言特徵 (句法、文法、情感、可讀性) 進行詞嵌入 (word embedding)，包括 TF-IDF 和 CV (count vectorizer)，得出比 BERT 和 CNN 更高的準確度 [Verma et al. 2021]。

C. Propagation-based methods: 近年來社交網站快速發展，造成假新聞傳播的問題，因此藉由分析用戶行為和資訊傳遞模式來預測新聞的真實程度。Tacchini et al. 假設在社交網站上與新聞貼文互動的用戶可以用來檢測新聞的真實性，並得出高準確率。同時發現大多數假新聞擁有更多的點贊 [Tacchini et al. 2017]。Vosoughi et al. 運用聯級 (cascade) 量化 twitter 的推文，去計算用戶數量、回推 (retweet) 次數、和結構性病毒擴散。分析結果發現，假新聞的聯級明顯大於真實新聞的聯級，涉及的用戶數、回推次數和病毒傳播也比真實新聞多，並且關於政治和都市傳說的假新聞傳播速度最快 [Vosoughi et al. 2018]。

D. Source-based methods: 我們可以透過評估新聞來源的可信度來判斷假新聞。Nørregaard et al. 整理了八個不同網站資料 (Ground true data) 並對每個來源的真實程度進行標記，統計結果顯示來自不可靠來源的新聞文章大多數是假新聞 [Nørregaard et al. 2018]。

我們主要針對新聞內容去做分析，因此主要會參考 style-based 的方法實作。

V. Methodology

Preprocessing 前處理：目標為減少文本量，減少差異性，讓文本達到最精簡卻不失其特徵。

方法: 包括 (1) 去除非文本，像是 html tag 之，以及標點符號和數字。(2) 修正拼寫錯誤。(3) 變形還原回單字原樣，像是 ing, 複數 s，過去分詞 ed。(4) 去掉 stopword，像是冠詞，be 動詞等對語意不會有太大影響的詞。(5) 斷詞，讓文本精簡並易於處理。

NLP 預訓練模型

Google BERT [1]

BERT，是 Bidirectional Encoder Representations 的縮寫，它從單詞的左右來考慮上下文。這種雙向性有助於模型更好地理解使用單詞的上下文。此外，BERT 的設計目標是能夠進行多任務學習，也就是說，它可以同時執行不同的自然語言處理任務。BERT 也是第一個無監督、深度雙向的自然語言處理模型預訓練系統。它只使用純文本語料庫進行訓練。模型來源: https://paperswithcode.com/paper/bert-pre-training-of-deep-bidirectional#code

- 利用以上預處理過的 model，可以省去訓練的成本，直接拿到結果 ，並且可以更加準確，也避免重複造輪子的行為。
- 將預處理過的文本應用到上述的 model 中，可以拿到對應的結果。
- 此方法能直接分好類並預測出結果，可以不需要使用到接下來會討論到的 classfication 的方法。

NLP analyze the texts：目標在於將把作為 sample 的每個文本轉換成 Feature vector，讓我們能使用分類器將其分類

Bigram

- 透過下方公式，算出根據前後文的出現機率，算出每個 feature pair 對應的 feature vector。

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})}$$

- 此方法快速，但前後文的依賴性不高，feature vector 能表達出來前後文的關係很有限。

Word2Vec - Skip-gram

- 從一個字詞，到產生這個字詞的詞向量，再到預測其他可能會在那個字詞附近出現的詞彙的機率。
- skip-gram 模型的訓練目標是最大化在給定目標詞的情況下預測上下文詞的概率。

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\top} v_{w_I}\right)} \qquad \frac{1}{T}\sum_{t=1}^{T} \sum_{-c\le j\le c, j\neq 0} \log p(w_{t+j}|w_t)$$

- 有點類似 Figure 1 做 convolution 的方式，可以同時運用到前後文來產生 feature vector，能有較高的依賴性
- 可以藉由上面的運作方法，將字串 list 轉成 feature vector。

Classifiers 分類器：目標在於根據上一步驟拿到的 feature vector，以及 database 中的 vector，運用監督式學習的模型，進行假新聞的預測。

Decision tree

- 運用 feature vector，使用 decision tree 的演算法，可以訓練出 tree。
- 此方法快速且對於只要分兩類的問題，通常很有效果。
- 如果覺得精準度不足，可以升級為 decision forest，運用 polling 的方式，來增加其精準度。
- 但是對於 feature vector 的運用稍嫌不足，且長期依賴性不高。

Adaboost

- 前一個基本分類器分錯的樣本會得到加強，加權後的全體樣本再次被用來訓練下一個基本分類器。同時，在每一輪中加入一個新的弱分類器，直到達到某個預定的足夠小的錯誤率或達到預先指定的最大疊代次數，達到所謂自適應的分類器。
- 公式：
  - 首先，adaboost 為每個分類器給定一個權重值 α，這些 α 是由錯誤率進行計算而得。

$$\epsilon = \frac{\sum \text{未正确分类的样本的权值}}{\text{样本数目}} \qquad \alpha = \frac{1}{2} ln(\frac{1-\epsilon}{\epsilon})$$

  - 而訓練資料中的每一個樣本，賦予一個權重，這些權重構成了向量 D。

$$D_i^{(t+1)} = \frac{D_i^{(t)} e^{-predict*label*\alpha}}{Sum(D)}$$

- AdaBoost 把多個不同的決策樹用一種非隨機的方式組合起來，表現出驚人的性能！
  - 把決策樹的準確率大大提高，可以與 SVM 媲美。
  - 速度快，且基本不用調參數。
  - 幾乎不 Overfitting。

LSTM

- LSTM 使用記憶來加強當前的決策，利用三個控制閥來決定記憶的儲存與使用
  - 遺忘閥 :如果當前的字句是新主題或以前面字句相反的詞，那麼，之前的字句就會這個閥過濾掉，反之，可能就會被繼續保留到記憶中。
  - 輸入閥 (以 it 表示)：決定當前的輸入及新產生的記憶單元是否加入長期記憶中。
  - 輸出閥：決定當前的字句是否加到輸出，這個閥也是 Sigmoid 函數，表示要加入與否。
- 這個方法雖然要使用的資源看似最多，但她可以解決前兩個方法的一個大問題:長期依賴性不高。因為它會自行判斷在甚麼時候捨棄前文。

VI. Evaluation Metrics

Accuracy

Precision：回傳資料的精準度。

$$Precision = \frac{TP}{TP + FP}$$

Recall：回傳的正確資料佔所有應該被回傳的資料比

$$Recall = \frac{TP}{TP + FN}$$

F1-score：1:1 權衡 Precision & Recall

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

MCC (Matthews Correlation Coefficient)：避免資料集偏度過高，導致對分類器的評價失準

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Loss rate : 算出 dataset label 與分類的差異。

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

VII. Baselines

- 實作出 N-gram + Decision Tree
- F1 score 達 80 分

VIII.Work Plan

Time schedule:

- Data Preprocessing
- Style-based analysis: (1) bigram, (2) word2vec
- Classifers: (1) BERT, (2) Decision tree, (3) Adaboost (4) LSTM
- Evalution

Discussion: https://docs.google.com/document/d/18Npon0XLoFYIaJjk2bpSagqGhIGTyTumxrvYBMi1OCg/edit#

Repo link: https://github.com/dianel0922/Final-Project

IX. Reference

1. https://kknews.cc/code/ey94pkr.html
2. https://www.tensorflow.org/tutorials/text/word2vec
3. https://www.itread01.com/content/1547087057.htm
4. https://docs.google.com/presentation/d/1UHXrKL1oTdgMLoAHHPfMM_srDO0BCyJXPmhe4DNh_G8/pub?start=false&loop=false&delayms=3000&slide=id.g24de73a70b_0_0
5. https://medium.com/ai%E5%8F%8D%E6%96%97%E5%9F%8E/evaluation-metrics-%E5%88%86%E9%A1%9E%E6%A8%A1%E5%9E%8B-ba17ad826599
6. Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys (CSUR)*, *53*(5), 1-40.
7. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter, 19(1), 22-36.
8. Zhang, X., & Ghorbani, A. A. (2020). An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management, 57(2), 102025.
9. Horne, B., & Adali, S. (2017, May). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In Proceedings of the international AAAI conference on web and social media (Vol. 11, No. 1, pp. 759-766).
10. Wynne, H. E., & Wint, Z. Z. (2019, December). Content based fake news detection using n-gram models. In Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (pp. 669-673).
11. Li, Q., Hu, Q., Lu, Y., Yang, Y., & Cheng, J. (2020). Multi-level word features based on CNN for fake news detection in cultural communication. *Personal and Ubiquitous Computing*, *24*(2), 259-272.
12. Verma, P. K., Agrawal, P., Amorim, I., & Prodan, R. (2021). WELFake: word embedding over linguistic features for fake news detection. IEEE Transactions on Computational Social Systems, 8(4), 881-893.
13. Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506.
14. Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. Science, 359(6380), 1146-1151.
15. Nørregaard, J., Horne, B. D., & Adalı, S. (2019, July). NELA-GT-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In Proceedings of the international AAAI conference on web and social media (Vol. 13, pp. 630-638).

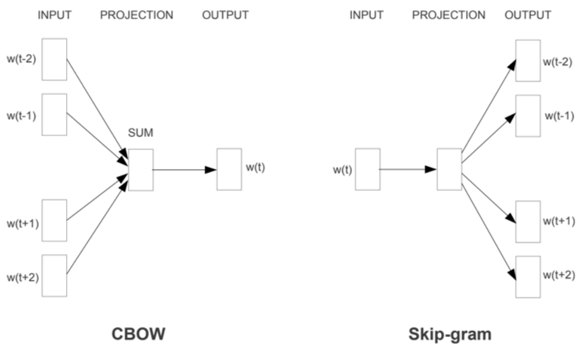


Figure 1: Word2Vec - Skip-gram

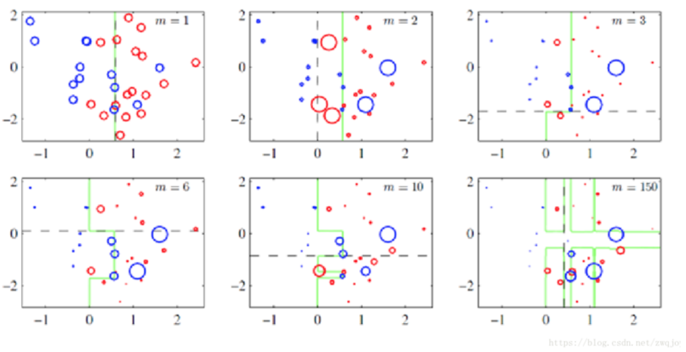
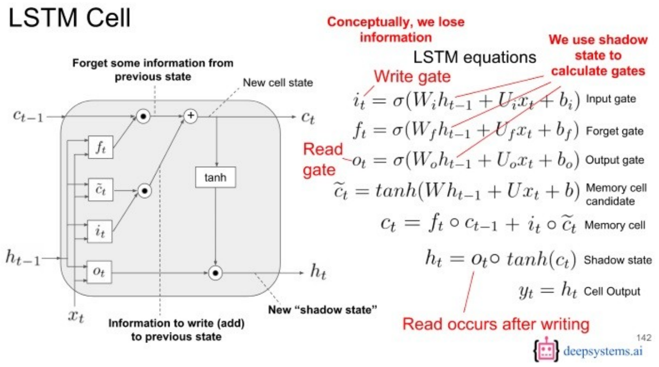


Figure 2: Adaboost



LSTM Cell

Forget some information from previous state

New cell state

Information to write (add) to previous state

New "shadow state"

Conceptually, we lose information

LSTM equations

We use shadow state to calculate gates

Write gate

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$ Input gate

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$$ Forget gate

Read gate

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$ Output gate

$$\widetilde{c}_t = tanh(W h_{t-1} + U x_t + b)$$ Memory cell candidate

$$c_t = f_t \circ c_{t-1} + i_t \circ \widetilde{c}_t$$ Memory cell

$$h_t = o_t \circ tanh(c_t)$$ Shadow state

$$y_t = h_t$$ Cell Output

Read occurs after writing

142

deepsystems.ai

Figure 3: LSTM