國立台北大學資訊工程學系專題報告

# AI Laws Assistant for Intellectual Property Rights

# – AI 法務助手-智慧財產權

專題組員:邱奕銘、潘文傑、陳郁庭、蔡若穎

專題編號: PRJ-NTPUCSIE-106-001

執行期間:106 年 09 月 至 107 年 05 月

## 一、 摘要

早在 1950 年圖靈在他名為〈運算機器與智慧（Computing Machinery and Intelligence)〉的論文中提出了著名的「圖靈測試」:如果機器與人類進行非面對面（例如在中間以布幕隔離）對話（例如使用文字訊息)，人類卻無法辨認出對方是機器，那麼這台機器就具有智慧。

歷經了 60 年的發展,人工智慧的研究領域因種種困難而起起落落，經歷了無數個轉角。如機器無法應付計算複雜度的困境，然而在 2017 年可以說是人工智慧應用大爆發的一年，各式各樣的 AI 應用都在不同領域中有所突破，從交通、醫療到娛樂服務。比如:AlphaGo Zero 戰勝原版 AlphaGo，只需要三天就能熟悉第一代 AlphaGo 所有的圍棋知識；Google 旗下自駕車計畫公司 Waymo,撤去人類監督職位，成為全球首次「真」無人車上路測試；中國科大訊飛的一款 AI 機器人「小易」，通過了臨床執業醫師綜合筆試評測，成為全球第一位通過醫療執照考試的機器人；甚至在阿拉伯聯合大公國，出現了全球首位獲得公民權的 AI 機器人索菲亞（Sophia)。對人類來說，如何讓這些現代自己製造出來的機器們，可以聽懂人話，並與人類「合作」。

本專題研究將透過自然語言處理，讓機器理解人為語言，實現一個應用於智慧財產權的法律環境下，當使用者輸入相關案件或他可能犯罪的行為時，透過從巨量判決書中訓練出的神經網路模型，來預測使用者違法機率與提供過往相似度較高之判決。

## 二、 簡介

### （一） 研製背景

在人工智慧技術的發展下，法律逐漸走向智慧化以及自動化的方向發展。法律研究對於從業律師、公司法務、司法人員以及法學院學生都是必要的，甚至一般民眾都會需要進行法律諮詢(Legal advice)與法律檢索。但 Westlaw、司法院法學資料檢索系統等法律專業知識資料庫都是使用傳統的關鍵字檢索，使用傳統關鍵字檢索對於一般民眾缺乏相關法律背景知識是一件不智慧又費時費力的工作。

### (二) 研究目標

本專題導入深度學習技術至法律系統中,利用深度學習標記關鍵要素，結合深度學習之深度神經網路，藉由建立法學資料庫，透過斷詞系統將中文進行斷詞處理，並透過法律關鍵字自動標記出該判決書為勝訴或敗訴，再透過 AI 模型將經由斷詞處理後的判決書由深度學習準確判斷各判決書之

差異，期望訓練完成後的模型當有一筆新的判決書或是專業法律問題輸入時能準確預測勝訴敗訴。

### （三） 主要預期效益

1. AI 系統能對使用者之輸入進行準確地預測是否違法
2. 能準確提供使用者輸入之過往相似判決
3. 使用此系統能實際減少大量法務人員查詢過往判決書之流程
4. 製作出簡單易使用的使用者介面(User Interface)。
5. 透過專題實作，更了解人工智慧之相關應用。

## 三、 專題進行方式

### （一） 實作平台與技術

#### 1. 開發環境：

本系統開發環境如下圖：

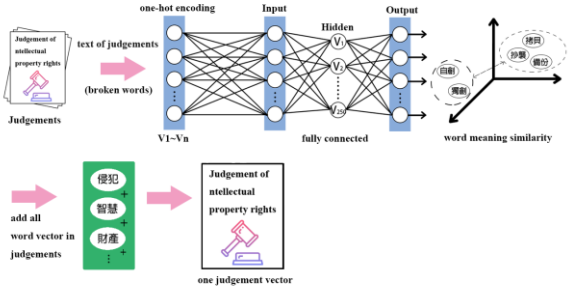



本系統架構分為三大部分：

**(a) 使用者端**

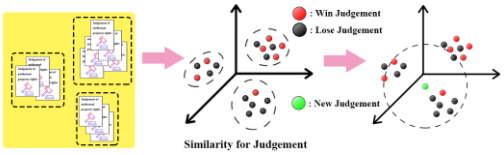
使用者端我們使用 RWD 網頁作為主要呈現方式並以 Angular 作為前端框架進行網頁系統的撰寫

**(b) AI 伺服器**




系統流程圖

此部份我們利用 gensim 其所提供之 Word2Vec 套件先對資料庫所有判決書內文中的詞彙進行訓練，訓練完畢後再將單一判決書中的所有詞的詞向量作加總用以得到代表該判決書的文本向量。




文本分群流程圖

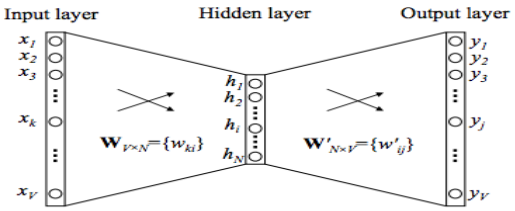
之後再以各文本向量的相似來對各判決書作分群。

資料庫我們以 MongoDB 作為我們的後端資料庫，並使用 python 作為 AI 模型建立之主要語言，之後再以 Flask 來實作後端 Restful API 之程式與前端 Angular 進行使用者請求與資料的交換

**(c) 資料抓取與處理**

使用 Selenium 撰寫爬蟲程式，至 Lawsnote 抓取過往大量的過往判決書，並透過 Jieba 之自定義斷詞系統對判決書內文進行斷詞後再作為 AI 的輸入。

#### 2. Word2Vec 介紹




上圖Word2vec之架構,左邊Input將判決書內的標籤詞輸入神經網路內，

經由訓練得出代表個標籤詞語意的詞向量(即為$W_{VxN}$)，得到內部各詞的詞向量後，進而將單一判決書內所有詞彙之詞向量進行向量加總，即可得到代表該判別書文本意含的文本向量，可模擬法務人員在搜尋相關行為是否觸法時，搜尋文章內容較為相似的文章去參考並判斷該行為是否觸法。

### 3. TF-IDF

在此探討在上述Word2Vec中得到各判決書的文本向量後，如何再進一步得到對各判決書文本含意的文本向量進行優化，各詞彙的順序、出現頻率皆會影響此篇文章所代表的含意，而若單純的加總詞向量就代表該文本的文本向量，並不能精確表達文本，在此，我們將使用 TF-IDF 來處理 1. 所得到之詞向量，藉此得到更精準的文本向量，TF-IDF 包含兩部分詞頻跟逆向文件頻率。

詞頻(TF)為考量詞所出現的頻率，而不是單從次數去考量，如此可解決當兩文本中該詞出現的次數相同，單純加總的話該詞對兩文本的影響度會相同，但在文本較長的文本中，該詞的涵義重要性較低，而在文本較短的文本中，該詞的涵義重要性就較高。

其計算方式為對於在某一特定檔案裡的詞語 $tf_{i,j}$ 來說，它的重要性可表示為

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

以上式子中$n_{i,j}$是該詞在檔案$d_j$中的出現次數，而分母則是在檔案 $d_j$ 中所有字詞的出現次數之和。

逆向文件頻率為解決常用詞(如:這個、的)的問題，若該詞出現頻率高且常出現在各文本中，則該詞涵義重要性就相對較低，但若該詞不常出現在各文本中，則該詞的涵義重要性對該文本來說就高。某一特定詞語的idf，可以由總檔案數目除以包含該詞語之檔案的數目，再將得到的商取對數得到：

$$\mathrm{idf_i} = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

其中

$|D|$：語料庫中的檔案總數

$|\{j : t_i \in d_j\}|$：包含詞語 $t_i$ 的檔案數目（即 $n_{i,j} \neq 0$ 的檔案數目）如果詞語不在資料中，就導致分母為零，因此一般情況下使用

$1 + |\{j : t_i \in d_j\}|$

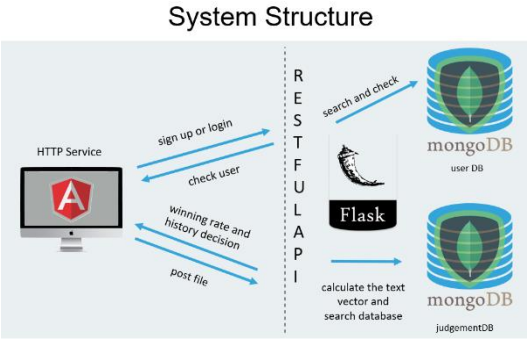
而最後

$$\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \times \mathrm{idf_i}$$

某一特定檔案內的高詞語頻率，以及該詞語在整個檔案集合中的低檔案頻率，可以產生出高權重的 tf-idf。因此，tf-idf 傾向於過濾掉常見的詞語，保留重要的詞語。

## （二） 系統分析與設計摘要

本系統架構圖如下




前端方面利用 Angular 來與使用者進行互動，當需要進行資料庫檢索與存取則透過 RestfulAPI 發送請求至後端 Flask 的程式中再藉由 Flask 對 MongoDB 進行操作

1. 登入

登入時由前端發送登入請求，後端檢查資料庫內的帳號與密碼是否正確後，回傳登入是否成功給前端頁面

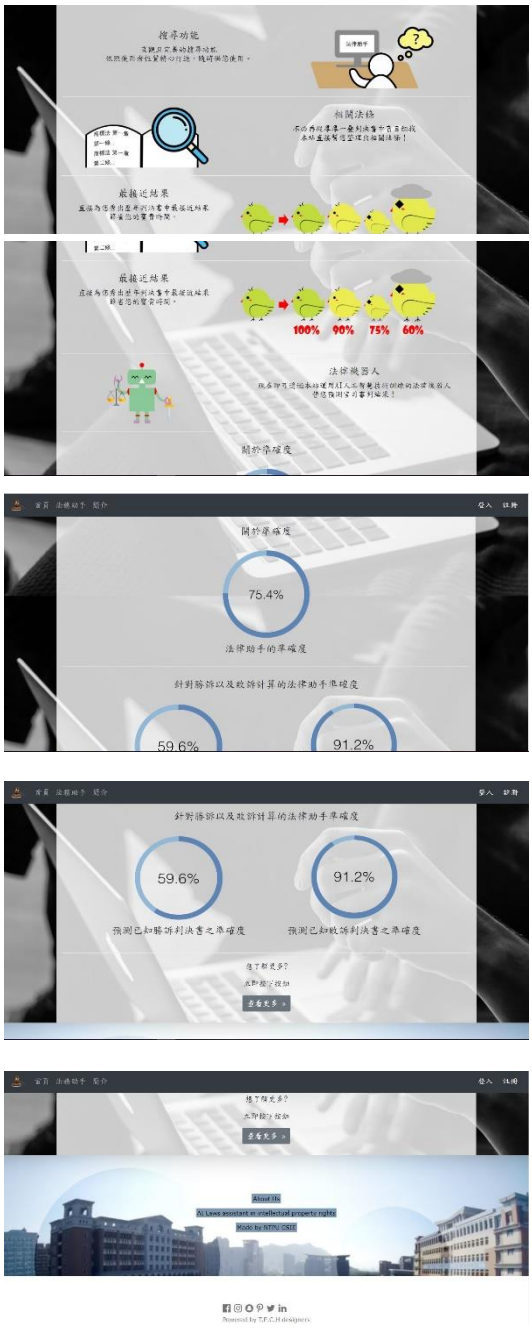
2. AI 智慧檢索

該功能主要分為輸入案件之文本檔案或關鍵字檢索，系統接受到使用者上傳的檔案或語句後，將該文本或語句分析並得出輸入之文本向量後進入系統之判決書資料庫進行檢索，找出最相近之判決書，並以相關度大小進行呈現的順序，再回傳給使用者相關的判決書。

# 四、 主要成果與評估

## （一） 網頁成果

### 1. 首頁


該頁面展示了整個網站所具備的功能，並簡單介紹網站特色，以及系統成果，在右上角使用者可進行註冊與登入會員，登入後的使用者才會使用我們的法務助手功能。

## 2. 簡介



該頁面詳細介紹本網站的功能該如何使用並以動態圖示範實際操作流程來讓使用者更清楚如何去使用本網站

## 3. 法務助手

如上圖法務助手方面提供 2 種搜索方式 1.上傳案件內文檔案進行類似案件搜尋 2.以關鍵語句進行搜索以前者方式進行搜索因文本內容較多，因此對於系統來說較能找到相似的過往判決書。

### (a.) 案件內文檔案檢索




使用者將案件內文檔案上傳後，即可得到經由系統計算過後的相關判決，並且回傳結果也以相似度進行排序，回傳結果的事由也大多與上傳案件的事由雷同，在分群這塊系統有成功做到此功能，使用者若想詳細觀看某一判決書也可直接點選看內文，即可得到該判決書的詳細資料。

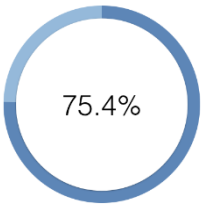
### (b.) 關鍵語句搜索



在本功能中使用者可透過關鍵語句進行檢索(例:抄襲他人圖畫並用於販售)，同 a.一樣會返回搜尋結果並可觀看個別判決書的詳細資料，在此功能因輸入的內容較少，因此和 a.明顯

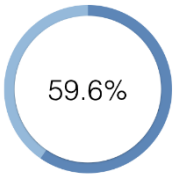
有相似度的差異，此功能較難找到相似度很高的過往判決，但返回判決也大多與輸入是有相關性的。

### （二） 準確程度




總準確度

將資料庫中已知為勝訴敗訴的判決書再次輸入至系統中讓系統判斷其偏向勝訴或敗訴，其結果的準確性為75%，再來測試方式又分成將已知是勝訴的判決書判斷正確機率與已知是敗訴的判決書判斷正確機率。




已知勝訴判決書判斷正確率


已知敗訴判決書判斷正確率

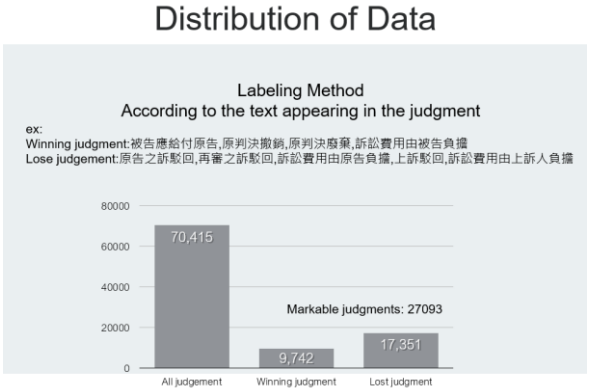
其中發現，已知為勝訴的判決書判斷正確機率相當低，而已知為敗訴的判決書判斷正確機率較高，推斷為資料庫中敗訴的判決書的比例較高，導致系統的判斷容易偏向於判斷其為敗訴。

### （三） 未來可擴展方向

#### （a.） 斷詞

由於在中文斷詞比英文斷詞更加困難，且斷詞成果也會直接影響系統的準度，若能更有效增加斷詞的準確性，對整體系統的精確度也將有很大的提升。

#### （b.） 勝敗訴判斷




資料庫分布圖

上圖顯示資料庫的分布，其中總資料有 70415 筆的判決書，以標記勝敗訴的有 27093 筆，其中勝訴為 9742 筆、敗訴為 17351 筆，勝敗訴的比例差距相當高，且總筆數與已判斷的筆數也是差距相當高，判斷勝負是採取，查詢內文中是否提及某些法律勝負關鍵字，若能增加關鍵字的數目，即可成功擴充可使用的資料，對系統效能將有很大的幫助。

## 五、 結語與展望

本次的專題研究中，成功了實踐了讓電腦自動分群的效果，雖然離在原先預期的預測判決上並沒有相當成功的效果，但本系統仍完成能提供法務人員快速搜索過往相似判決書的功能，能有效減少法務人員在翻閱過往判決書的時間，仍是一個能成功實用在社會上的系統，在專題過程中組員也學習到了許多，無論是團隊溝通、分工，以及整個專題的流程時間規劃，能力都有明顯的提升，對於如何生成一個產品，也有了更加深刻的體悟，也期望未來能將該系統開發得更完善，將 AI 成功應用於社會進步上。

## 六、 銘謝

特別感謝指導教授的指導以及提供資料給我們使用的網站。

## 七、 參考文獻

[1] 周秉誼，淺談 Deep Learning 原理及應用，http://www.cc.ntu.edu.tw/chinese/epaper/0038/20160920_3805.html

[2] Westlaw:https://www.westlaw.com/

[3] 北大法寶法律數據庫:www.pkulaw.cn/

[4] Lawsnote:https://www.docker.com/

[5] Goldberg, Yoav; Levy, Omar. word2vec Explained: Deriving Mikolov et al.’s Negative-Sampling Word-Embedding Method

[6] Rehurek, Radim. Word2vec and friends (Youtube video)

[7] http://blog.fukuball.com/ru-he-shi-yong-jieba-jie-ba-zhong-wen-fen-ci-cheng-shi/ 自定義斷詞資料庫

[8] Guangyi Xiao, Jiqian Mo, Even Chow, Hao Chen, Jingzhi Guo, and Zhiguo Gong, “Multi-Task CNN for Classification of Chinese Legal Questions,” *The Fourteenth IEEE International Conference on e-Business Engineering*, Shanghai, China, November 2017.