基於多 BERT 模型之 NLLP 應用於建築工程訴訟之理解與預測

NLLP for the Understanding and Prediction of Construction Litigation Based on Multiple BERT Model

鍾文傑 Wen-Chieh Chung, 陳哲文 Che-Wen Chen, 王駿發 Jhing-Fa Wang
國立成功大學電機工程學系
Department of Electrical Engineering, National Cheng Kung University
n26074883@gs.ncku.edu.tw, kfcmax300@gmail.com, wangjf@mail.ncku.edu.tw

曾世邦 Shih-Pang Tseng
常州信息職業技術學院軟件與大數據學院
Software Department, Changzhou College of Information Technology, Changzhou
tsp@tajen.edu.tw

王宗松 Tzong-Song Wang
大仁科技大學數位多媒體設計系
Department of Digital Multimedia Design, Tajen University
tswang@tajen.edu.tw

摘要

本研究以深度學習之 BERT 技術提出一個工程訴訟案件篩選與歷審統計表建立及案件預測系統，並分為三個部分。第一部份是工程訴訟案件篩選，由中華民國司法院提供之判決書資料中經由基於 BERT 的模型架構篩選出屬於建築工程訴訟之案件，其準確率達到 93.55%。第二部分是案件歷審統計表建立，將案件的歷審判決書利用正則表達式進行資訊擷取並彙整成個案之歷審統計表，準確率達到 86.75%。第三部分是案件預測，利用基於多 BERT 的模型架構預測法院判決之結果，並找出相似的案例及同案件類型之統計表格，而判決預測在金額上及時間上準確率分別達到 82.22%及 88.89%。

關鍵詞：案件篩選，資訊擷取，文本相似度，判決預測，BERT

Abstract

This research uses the multiple BERT model to propose an construction litigation case screening and summary table creation and case prediction system, which is divided into three parts. The first part is the screening of construction litigation cases. From the judgment data provided by the Judicial Court of the Republic of China, the cases belonging to the construction litigation are selected through the BERT based model structure, and the accuracy rate reaches 93.55%. The second part is summary table creation, which uses regular expressions to extract information from the judgments and integrate them into a case summary table, with an accuracy rate of 86.75%. The third part is the case prediction. The multiple BERT model framework is used to predict the outcome of court judgments, and to find similar cases and statistical tables of the same case type. The accuracy rates of judgment prediction in terms of amount and time are respectively 82.22% and 88.89%.

Keywords: Case Screening, Information Extraction, Text Similarity, Judgment Prediction, BERT

一、緒論

隨著人工智慧技術日新月異，各個領域也逐漸加入深度學習的技術以加速產業之發展，而在法律的領域也有相關應用，近年 NLP 在法律上的應用被稱為 Natural Legal Language Processing (NLLP)[1]，利用 NLP 技術有效且精確解決法律上的相關問題，解決以往需要大量基層人員花費多數工時在進行法律資訊檢索等工作。NLLP 相關應用像是基於法律文本的問答系統[2]，預測法院的投票與判決預測[3][4]，判斷案例的描述是否違反人權條款[5]和以單獨親權酌定裁判的預測模型為例[6]等。

在中華民國法院訴訟中，工程糾紛訴訟具有高度複雜性，且往往涉及之金額都非常龐大，訴訟所需時間也時常曠日廢時，而具有建築工程相關知識的法律專家更是少之又少，因此不論是法官、律師或是從事建築工程之人員都須參考過往之訴訟案件來解決紛爭。

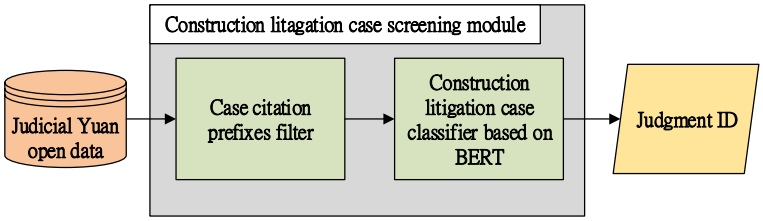
以現今中華民國司法院裁判書查詢系統提供之裁判書查詢方式僅以裁判書全文是

否出現某些關鍵字做為搜尋方法，雖能大致篩選出想查詢的相關裁判書資料，但其中可能混雜了一些與建築工程訴訟不相關的案例，且查詢到的資料對於工程法律缺乏全面性的整理與統計分析，使得在查找相關案例時必須花費大量的時間。

為了有效解決以往花費大量人力及時間在法律資訊檢索上，以及在訴訟前可以有效的評估訴訟效率，本系統將分為三個部分。第一部分是從中華民國司法院公開之裁判書資料中篩選出屬於建築工程訴訟之案件。第二部分是將屬於建築工程訴訟之案件利用正則表達式對判決書進行資料擷取並整理成案件歷審統計表格。第三部分是對新的案件利用 BERT 模型進行判決結果的預測，並找出相似之案例及同類型案例的統計表。

二、工程訴訟案件之案件篩選

案件篩選將從中華民國司法院提供之裁判書中篩選出屬於建築工程訴訟的案件，此模組會先利用案件的貫字進行第一次篩選，再透過基於 BERT 架構的網路模型進行第二次的篩選，取出其中屬於建築工程訴訟的案件。



圖一、工程訴訟案件篩選架構圖

（一）案件貫字篩選

在利用神經網路篩選工程訴訟案例前，我們希望用簡單且快速的方法過濾掉一些案件，進而節省神經網路分類案件的時間，而從案件的貫字可以大致知道案件的類型，我們從中刪除確定不屬於工程訴訟的案件，確定不屬於工程訴訟案件的貫字如表一，此為案件貫字的連結。

表一、刪除之案件貫字

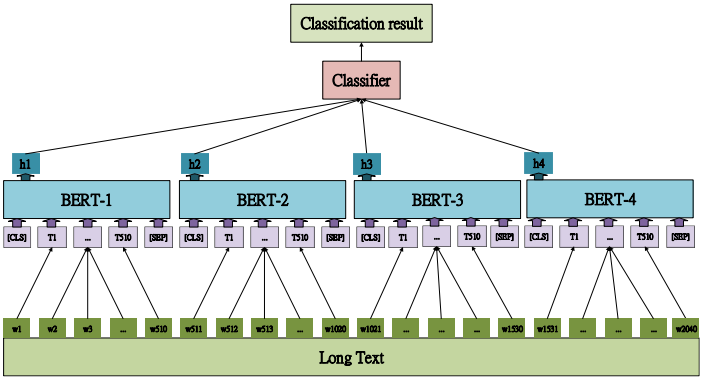
案件貫字	描述
婚	婚姻事件
家	家事訴訟事件
繼	繼承事件
醫	醫療糾紛事件

海	海商事件
國貿	國際貿易事件
金	證券、金融事件
選	選舉訴訟事件
親	親子關係事件
拍	拍賣事件
除	除權事件
智	智慧財產事件
聲	聲請、聲明事件
簡	簡易事件(金額 50 萬以下)

（二）基於 BERT 模型建構之工程訴訟案件分類器

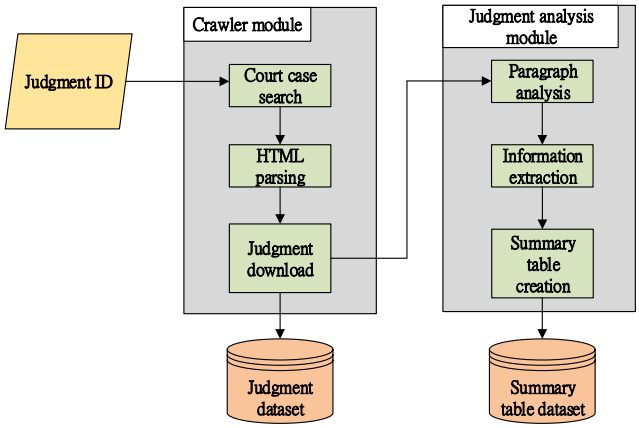
在刪除確定不屬於工程訴訟的案件後，剩餘的案件會透過由 BERT 模型[7]構建的分類器來進一步篩選，由於 BERT 模型有輸入文字長度的限制，故本系統中會使用 BERT 模型對切分成多段的輸入文本進行文本嵌入，再將多個 BERT 輸出的向量串接後進行後續的分類任務。

BERT 模型的輸入序列長度限制為 512 個字，扣除輸入序列最前面的***[CLS]***及最後的***[SEP]***後，最多可以輸入 510 個中文字，而在法院判決書中往往是數千字的文本，若由只保留文本前 512 字或文本後 512 字會損失大量的訊息，無法達到準確分類的效果。在本系統中將取出判決書中前 2040 個字，並拆分成四段後分別送入 BERT 模型中進行分類任務的模型微調，而四個 BERT 模型的參數為共享的，在模型訓練時，我們微調 BERT 最後四層的參數，其輸出的 4 個 768 維的向量串接後再進行下游的分類任務，此分類器的目的在於分類出屬於建築工程訴訟的案件與非建築工程訴訟的案件。



圖二、基於 BERT 架構之分類器

三、工程訴訟案件之歷審統計表建立



圖三、工程訴訟案件歷審統計表建立之架構圖

此部分包含兩個模組:(1)網路爬蟲模組用於自動化至中華民國司法院的裁判書查詢系統將指定的判決書下載，並儲存至判決書資料庫中。(2)判決書分析模組用於分析下載下來的判決書，其中會對判決書進行段落分析再提取各段落中重要的資訊，最後生成該案件的歷審統計表。

（一）爬蟲模組

1. 案件判決書搜尋

案件判決書搜尋是使用 python 的 selenium 套件至中華民國司法院的裁判書查詢系統(https://law.judicial.gov.tw/FJUD/default.aspx)自動化鍵入判決書 ID，並透過比對法院之區域與裁判書年份及編號，以確定搜索到正確的裁判書。

2. HTML 解析

HTML 解析是利用 python 的 beautiful soup 套件將搜索到的判決書網頁進行 HTML 解析，轉換為結構化的 beautiful soup 物件，可藉由指定 HTML 標籤的方式來快速找到判決書內文。

3. 歷審判決書下載

案件的歷審裁判在個案判決書的網頁中，可以透過 HTML 解析取得歷審裁判書的連結網址再透過解析歷審判決書網頁找到判決書內文並將歷審判決書內文下載儲存至判決書資料庫。

（二） 判決書分析模組

1. 段落分析

在本系統中，建築工程訴訟的判決書內文可分為 5 個段落:(1)原、被告資訊(2)主文(3)原告主張(4)被告抗辯(5)法官判決。

(1) 原、被告資訊描述了原告及被告雙方當事人及雙方法定代理人及訴訟代理人，主要位於「主文」二字以上的段落。

(2) 主文部分簡要描述了法院的判決結果，包含賠償金額與訴訟費用的分配，主要位於「主文」二字以下至「事實及理由」以上的段落。

(3) 原告主張描述了原告方在訴訟中提出的聲明、主張，主要位於「原告主張」以下至「被告抗辯」以上的段落，依據書記官的風格可能使用不同的詞彙「原告起訴主張、原告之主張、原告方面、原告之聲明及陳述」等。

(4) 被告抗辯描述了被告方在訴訟中對於原告論述的辯駁，主要位於「被告抗辯」以下至「心證之理由」以上的段落，同樣詞彙像是「被告則以、被告之答辯、被告方面、被告之聲明及陳述」等。

(5) 法官判決部分描述了法官的想法與做出決斷的理由，主要位於「心證之理由」以下的段落，相似詞彙如「心證理由、法院之判斷、本院之判斷、兩造之爭執、兩造不爭執」等。

2. 資訊擷取

我們將從上個段落分出的判決書的 5 個段落裡以及司法院網站中提取 12 項資訊:(1)原告當事人(2)原告訴訟代理人(3)被告當事人(4)被告訴訟代理人(5)工程標的(6)契約價金(7)原告索賠金額(8)原告訴求項目(9)歷審判決金額(10)歷審訴訟費用(11)歷審字號(12)訴訟期程。其中第 1 項到第 10 項可以從判決書中提取，第 11 項及第 12 項則要從司法院網站中提取。

本系統中用於資訊提取的方法為正則表達式，正則表達式是電腦科學的一個概念，使用單個字串來描述或匹配一系列符合制定的句法規則之字串，在很多文字編輯器中，常用正則表達式來檢索、替換符合規則的文字。

下面將逐項說明 12 項資訊提取的方法:

(1) 原告當事人位於原、被告資訊段落，找到「原告」二字後的公司或機關，即為原告當事人。

(2) 原告訴訟代理人位於原、被告資訊段落，找到「被告」二字前且在「訴訟代理人」後的姓名即為原告訴訟代理人。

(3) 被告當事人位於原、被告資訊段落，找到「被告」二字後的公司或機關，即為被告當事人。

(4) 被告訴訟代理人位於原、被告資訊段落，找到「被告」二字後且在「訴訟代理人」後的姓名即為被告訴訟代理人。

(5) 工程標的是當事人雙方發生糾紛的工程案，通常位於原告主張段落，「系爭工程」前的工程名稱即為工程標的。

(6) 契約價金為當事人雙方所簽訂之合約金額，通常位於原告主張段落，「契約金額、契約總價、契約價金」後的金額即為契約價金。

(7) 原告索賠金額為原告向法院請求被告給付之金額，位於原告主張段落，「聲明」後的「被告應給付原告○○○元」即為原告索賠金額。

(8) 原告訴求項目為原告向法院請求被告給付之各個項目，位於原告主張段落，由於各個訴求項目之金額加總為原告索賠金額，故此部分會將原告主張部分的所有金額以正則表達式找出，並在刪除大於原告索賠金額的項目後以窮舉法排列所有組合，並找出加總等於原告索賠金額之組合，而含有這些金額的句子極有可能包含原告訴求項目。

(9) 歷審判決金額為各審法院判決之金額，位於主文段落，「被告應給付原告○○○元」即為法院判決之金額。

(10) 歷審訴訟費用為各審原告與被告要負擔之訴訟費用，位於主文段落，「訴訟費用由○○負擔」或「訴訟費用由○○負擔○分之○，其餘由○○負擔」，訴訟費用除了要判斷由哪一方負擔外，還需要計算負擔的比例。

(11) 歷審字號為案例的各審裁判字號，在司法院網站裁判書查詢系統中個案裁判書頁面的歷審裁判欄位，其中分為裁定與判決，但由於裁定主要用於與訴訟程序事項

相關的程序裁判，與案件事實較無關係，故只擷取判決部分。

(12) 訴訟期程為案例起訴至結案所花費的天數，在司法院網站裁判書查詢系統中個案裁判書頁面的歷審裁判欄位，當中記錄了歷審判決的日期，而起訴日期並未標明，故以第一審裁判日期往前算 180 天做為預估的起訴日期。

3. 歷審統計表建立

歷審統計表建立是將上節擷取到的資訊統整成表格，並對一些歷審的資訊數值的計算，如各審裁判費用的總和及判決比例的計算。生成的歷審統計表如圖四。

臺北地方法院 99 年建字第□號

1.當事人

關係人		一審	二審	三審
原告	□營造股份有限公司	原告		
	律師	陳□芳律師	陳□芳律師	陳□芳律師
被告	□市政府工務局衛生下水道工程處	被告		
	律師	陳□玲律師	陳□玲律師 陳□安律師 洪□彬律師	陳□玲律師

2.工程標的:第七期分管網工程第十標（□區天和公園附近地區）

3.契約價金:1 億 1,080 萬元

4.索賠金額:19125239

5.歷審字號:

(1)第一審：臺灣臺北地方法院 99 年度建字第□號判決

(2)第二審：臺灣高等法院 102 年度建上字第□號判決

(3)第三審：最高法院 104 年度台上字第□號判決

(a)

6.訴訟效率

原告訴求		一審判決	二審判決	三審判決	確定判決	
					金額	比例
索賠金額合計	19125239	15736113	1271939	1271939	1271939	6.7%
原告訴訟費用	訴訟費	-168321	-252481	-270516	-691318	
	律師費	-80000	-80000	-80000	-240000	
	計	-248321	-332481	-350516	-931318	
原告訴訟所得					340621	1.8%
期程	起(上)訴日期		101.12.30	103.5.6		
	判決日期	101.12.10	103.04.16	104.03.19		
	合計天數	1010 天				

原告訴求:

[14382911, ', 處原告逾期罰款 14,382,911 元', ', 又逾期罰款 14,382,911 元']

[4317526, ', 被告應給付展延工期之工程管理費 4,317,526 元']

[90000, ', 支出保險費 9 萬元']

[334802, ', 依法定利率計算為 334,802 元']

一審判決細項:

[14382911, ', 處原告逾期罰款 14,382,911 元', ', 又逾期罰款 14,382,911 元', ', 工期逾期 138 日應罰款 14,382,911 元', ', 原告依系爭契約第 9 條第 1 項第 6 款、第 12 條第 3 項約定及民法第 179 條規定請求被告返還 14,382,911 元', ', 原告得依系爭契約第 9 條第 1 項第 6 款、第 12 條第 3 項約定及民法第 179 條規定請求被告返還 14,382,911 元']

[1353202, ', 原告依系爭契約第 11 條第 5 項約定得請求展延工期之工程管理費 1,353,202 元', ', 另得依系爭契約第 11 條第 5 項約定請求被告給付展延工期之工程管理費 1,353,202 元']

(b)

圖四、系統生成的個案歷審統計表

原告訴求部分除了將金額找出外，同時也會擷取出包含金額的句子，減少人工找查判決書中原告訴求項目判決書的時間，如圖四(b)。

四、工程訴訟案件之案件預測

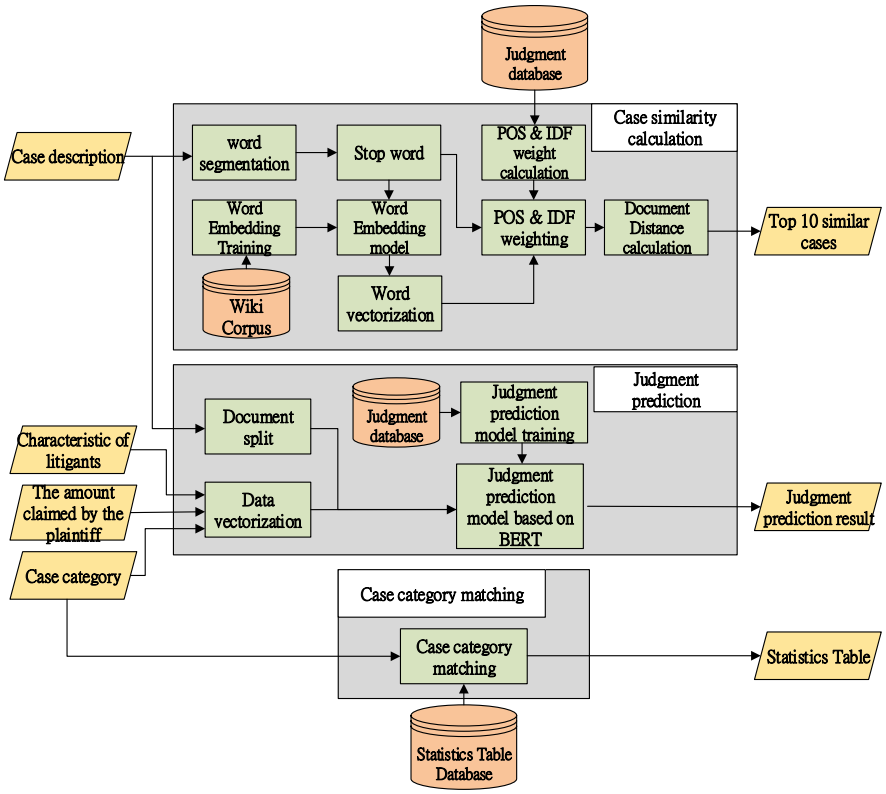
本系統預測部分分為三個功能，案例相似度計算、法院判決預測及案件類別匹配，本章節將依序介紹這三個功能，其架構圖如圖五所示。

（一）案例相似度計算

1. 中文斷詞及刪除停用詞

中文與英文在自然語言處理中最大的差異就是英文的每個詞在句子中都以空格分開，而在處理中文文本時，往往都需要先進行斷詞的處理，將句子拆分成更小的單位，以利於在後續處理中保留句子的完整意義。

在本系統中使用的斷詞方式是 CKIP Lab 所開發的斷詞系統[8]，相較於 jieba 斷詞在中文斷詞的表現上更準確。在做完中文斷詞後，接著，我們會進行刪除停用詞，停用詞主要是頻繁會出現的詞彙且即使刪除也可以表達句子意義的詞。



圖五、工程訴訟案件預測架構圖

2. 詞嵌入

經過前面的斷詞即刪除停用詞處理後，系統仍然無法理解每個字的涵義，所以我們必須將文本轉成向量的形式。本系統在案件相似度計算部分的向量化方式使用 word2vec[9]，其方法採用 Skip-gram 和 CBOW 兩種方法來訓練模型，將詞語映射到同一座標空間，其目的是讓相似上下文的詞會產生相似的詞嵌入結果。

3. 詞性(POS)與逆向文件頻率(IDF)加權

在將文本單詞向量化後，我們使用 POS 和 IDF 對向量進行加權。詞性標註的結果分為名詞、動詞、形容詞、副詞及其他。詞性的權重列於表二，我們起初設定名詞、動詞的權重為 5，其他為 1，再經過實驗調整後，得到形容詞、副詞的權重為 4 時會有最好的結果。

表二、詞性權重

詞性	權重
名詞、動詞	5
形容詞、副詞	4
其他	1

IDF 權重用於衡量單詞在文本普遍重要性的度量，其計算式如式(1)，idf_i為詞語t_i的 IDF 權重，$|D|$為語料庫中的檔案總數，j為包含詞語t_i的檔案數目，其中分母加一為避免除以零的情況。

$$idf_i = \log \frac{|D|}{|\{j:t_i \in d_j\}|+1} \quad (1)$$

結合 POS 和 IDF 權重的文本向量化表示式如式(2)。其中w_i表示字詞經 Word2vec 向量化結果，Pos_weight為根據每個詞的詞性所賦予的權重，Pos_{w_i}則是字詞經由 POS tagging 轉換的詞性，將文本內單詞分別經過 POS 和 IDF 的加權後相加，使每個文本都可以用 300 維的向量表示。

$$V = \sum_i w_i \times \log \frac{|D|}{|\{j:t_i \in d_j\}|+1} \times Pos\_weight(Pos_{w_i}) \quad (2)$$

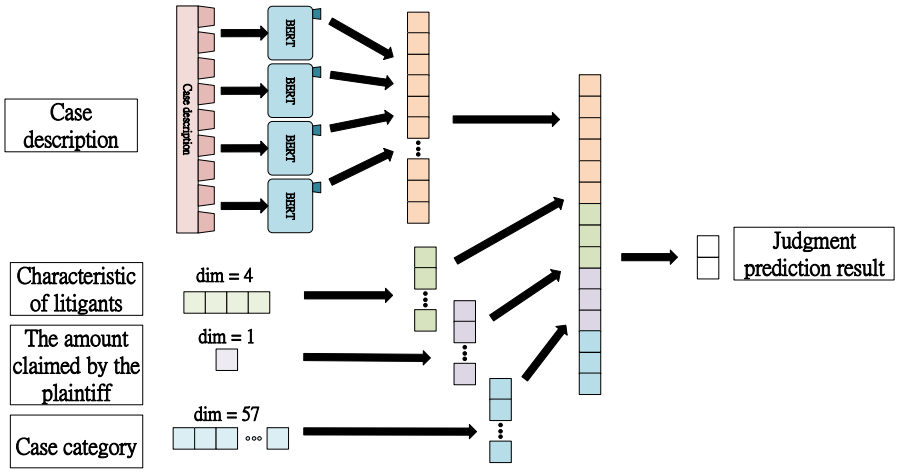
4. 相似度計算

在本系統中我們使用餘弦相似度來衡量文本之間的相似程度。餘弦相似度式通過測量兩向量夾角的餘弦值來度量他們之間的相似度。

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (3)$$

（二） 法院判決預測

判決預測的架構圖如圖六。判決預測模型的輸入包括(1)案件描述，即判決書中原告主張部分。(2)雙方單位性質，即公部門或私部門。(3)索賠金額。(4)案件類別，本系統中案件類別共 57 類。



圖六、判決預測模型架構圖

案件描述將取前 2040 個字做為輸入，將 2040 個字拆分成 4 個段落，每段最多 510 個字，並利用 BERT 模型進行嵌入，將 BERT 模型得到的 4 個 768 維向量串接。雙方部門性質因為雙方都有可能是公部門或是私部門，因此共有 4 種組合，並以 one-hot 的形式表示，故其維度為 4。索賠金額由於數值過大，我們會將輸入的索賠金額除以資料庫案件中最大的索賠金額。案件類別共有 57 類，以 one-hot 表示為 57 維向量做為輸入之一。

除了案件描述的其他三個輸入會經由全連接層提取 64 維特徵向量，並與 BERT 模型得到的向量串接再進行分類，兩個輸出維度為訴訟期間是否超過三年及金額上的勝敗訴。

（三） 案件類型匹配

案件類型匹配是將輸入系統的案件類型與 57 個案件類型做匹配，並輸出由建築工程訴訟方面的法律專家依照案件類型進行統計的統計表格。

五、實驗結果

（一） 案件篩選實驗

1. 案件篩選之資料庫

用於案件篩選的資料是從司法院裁判書開放資料網站(http://data.judicial.gov.tw/)所下載之 2000 年至 2018 年的裁判書，其資料庫內容如表 1，我們在 6,398,460 件民事案件中挑選了 1,231 件案件進行工程訴訟案件的標記，其中有 828 件為工程訴訟案件，403

件不是工程訴訟案件，而訓練集與測試集的切分比例為 9:1。

表三、案件篩選資料庫內容範例

項目	內容
裁判 ID	TPDV,88,訴,384,20000222
裁判年分	88
裁判貫字	訴
裁判編號	384
裁判日期	20000222
裁判案由	給付工程款
判決書內文	臺灣臺北地方法院民事判決 八十八度訴字第三八四號…… 事實 甲、原告方面：壹、聲明：除假執行擔保金額外，如主文所示。……

2. 案件篩選之實驗結果

我們使用第二章提到的模型進行工程訴訟案件的分類，並與其他方法進行比較，其中 keyword 是利用文本中出現特定關鍵字即將案件列為工程訴訟之案件，Word2vec+CNN 則是利用 Word2vec 將斷詞後的裁判書文本向量化並利用 CNN 作為分類模型，而 BERT 與 BERT-fine tune 則是未進行微調與進行微調的模型。結果如表四。

表四、案件篩選實驗結果比較表

方法	準確率
Keyword	62.67%
Word2vec+CNN	84.55%
BERT	91.86%
BERT-fine tune	93.55%

（二） 資訊擷取實驗

在資訊擷取實驗中，我們將系統產生出的 100 個案例的 word 檔與人工整理的 word 內容做比較，比較的內容包含(1)原告當事人(2)原告訴訟代理人(3)被告當事人(4)被告訴訟代理人(5)工程標的(6)契約價金(7)原告索賠金額(8)原告訴求項目(9)歷審判決金額(10)歷審訴訟費用(11)歷審字號(12)訴訟期程等 12 個項目。，正確率計算公式如式(4)。計算其每個案件的準確率後，取 100 件之平均準確率為 86.75%。

$$Accuracy = \frac{Wrong\ number\ of\ items}{Number\ of\ items} \times 100\% \quad (4)$$

（三） 案件相似度計算實驗

1. 案件相似度計算之資料庫

案例相似度計算的資料庫是由建築工程背景之法律專家所收集並整理 899 件工程訴訟案件，並針對個案分析其類別、雙方當事人、工程標的、索賠金額等資訊及判決書內文，本系統之案件相似度計算模組主要使用個案判決書中原告主張部分來比較案件之間的相似程度，並針對個案的案件類別進行各個方法的評比。

2. 案件相似度計算之實驗結果

將 899 件案件依次作為輸入案件計算案件相似度，並將每次的前 10 個最相似的案件之案件類別與輸入的案件類別比對，若輸出之案件的案件類別至少與輸入之案件的案件類別有一項相同，表示兩案之間有一定相關性，在前 10 個最相似的案件計算相關的案件數，並平均 899 件案件的相關案件數，其平均相關案件數如表五所示。

表五、案件相似度計算實驗結果比較表

方法	Average precision in TOP 10(AP@10)
TF-IDF	0.761
Word2vec	0.835
POS tagging + Word2vec	0.902
IDF + Word2vec	0.894
TF-IDF + Word2vec	0.866
POS tagging + IDF + Word2vec	0.911

（四） 法院判決預測

1. 判決預測之資料庫

判決預測使用的資料庫與案例相似度計算的資料庫相同，而在判決預測時要用到的部分是原告主張、雙方部門性質、原告索賠金額、案件類別，而訓練集與測試集的比例為 9:1。

2. 判決預測實驗結果

判決預測分為時間上勝敗訴及金額上勝敗訴，時間上勝敗訴以訴訟期程 3 年做區分，訴訟期程在 3 年以下為時間上勝訴，3 年以上為時間上敗訴，金額上勝敗訴則以訴訟所得佔據原告索賠金額之百分比為勝敗訴之依據，0%~25%為慘敗、25%~50%為小敗、

50%~75%為小勝、75%~100%為大勝，又可依 50%為分界，50%以上為勝訴，50%以下為敗訴，實驗結果如表六。

表六、判決預測實驗結果比較表

	方法	準確率
時間上	Word2vec+CNN	84.1%
	BERT	86.67%
	BERT-fine tune	88.89%
金額上(4 分類)	Word2vec+CNN	58.88%
	BERT	60.67%
	BERT-fine tune	64.44%
金額上(2 分類)	Word2vec+CNN	73.33%
	BERT	74.45%
	BERT-fine tune	82.22%

六、結論

本系統利用案件篩選及案件統計表建立兩個部分來輔助法律專家蒐集資料，解決了以往在法律資訊檢索上需花費大量人力及時間的問題。判決預測部分讓當事人或律師在訴訟前可以有效的評估訴訟效率，在評估是否進行法律訴訟。並可以針對相似的案例進行研究也有助於訴訟效率的提升。

在我們的系統中，先用貫字進行第一次的案件篩選，接著使用 BERT 模型來準確的分類屬於建築訴訟的案件，以避免神經網路所帶來的巨大運算時間以及傳統方法的低準確率問題，我們的分類器達到了 93.55%的準確率。在統計表建立部分使用正則表達式快速地對判決書進行分析，找出其重點部分並整理成表格，相較於以往人工整理表格大約可減少 6 倍以上的時間且準確率達到 86.75%，與本研究合作的建築師事務所之負責人表示，此系統預估可減少 30%的人力於此法律資訊檢索工作，節省花費的時間成本約每年 50 萬元。判決預測部分使用 BERT 模型對新的案件進行判決的預測，針對金額與時間進行分析，當事人可自行評估是否有訴訟的價值，而我們的模型在金額上及時間上分別得到 82.22%與 88.89%的準確率。此外，透過 POS 及 IDF 加權的詞向量進行相似案例的計算，則可讓律師對相似的案例深入研究以提高訴訟的勝率。

參考文獻

[1] Aletras, Nikolaos, et al. "Proceedings of the Natural Legal Language Processing Workshop 2019." Proceedings of the Natural Legal Language Processing Workshop 2019. 2019.

[2] Do, Phong-Khac, et al. "Legal question answering using ranking SVM and deep convolutional neural network." arXiv preprint arXiv:1703.05320 (2017).

[3] Katz, Daniel Martin, Michael J. Bommarito, and Josh Blackman II. "A general approach for predicting the behavior of the Supreme Court of the United States." PloS one 12.4 (2017).

[4] Virtucio, Michael Benedict L., et al. "Predicting decisions of the philippine supreme court using natural language processing and machine learning." 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). Vol. 2. IEEE, 2018.

[5] Aletras, Nikolaos, et al. "Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective." PeerJ Computer Science 2 (2016): e93.

[6] 黃詩淳, and 邵軒磊. "人工智慧與法律資料分析之方法與應用: 以單獨親權酌定裁判的預測模型為例." 臺大法學論叢 48.4 (2019): 2023-2073.

[7] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[8] Ma, Wei-Yun, and Keh-Jiann Chen. "Design of CKIP Chinese word segmentation system." *Chinese and Oriental Languages Information Processing Society* 14.3 (2005): 235-249.

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