

# 我國簡易刑事判決的製作輔助系統

## Decision Support for Criminal Summary Judgment

張正宗 何君豪\* 劉昭麟

國立政治大學 資訊科學研究所

板橋地方法院\*

{g9005, g8905, chaolin}@cs.nccu.edu.tw

### 摘要<sup>†</sup>

相對於國外，在台灣，資訊科學與法律的結合，一直很少有人研究，但是隨著時代的發展、科技的進步，各行各業紛紛引用最新的技術來處理面臨到的問題，甚至從事更新技術的研發。法律案件處理上所使用的技術，當然也不能停滯不前，必須跟進時代的腳步才是。針對於此，我們建立了一套輔助法官製作判決的系統，我們利用法律條文以及判決書，建立了輔助判決的 rule database，並以 Longest Common Subsequence 這一類問題的解法，來處理 rule 與起訴書中犯罪事實欄位間，中文語意的比對問題。利用此系統，我們藉由起訴書中犯罪事實欄位的資訊，找出該案件之案由，並提供應適用法條之建議。在我們的測試中，系統所建議之應適用法條的正確率達 90% 以上。

**關鍵字：**法學資訊系統，自然語言處理，Longest Common Subsequence

### 1. 概論

隨著網際網路的發展，各個機關的網頁紛紛出籠：個人網頁、學校網頁、公司的網頁，甚至是政府部門的網頁也不例外，大部份我們所需要的資料，都可在網路上取得，法律方面的資料也是一樣，司法院的法學資料全文檢索[12]，提供了中央與地方的法規查詢，以及司法解釋和法院的判例；法務部主管法規資料庫中[13]，除了提供中央法律的查詢外，大法官解釋、刑事判例、法律問題座談等資料，也都收錄在此資料庫中；法務部全國法規資料庫[14]，提供依各式法規類別查詢之服務，以及中外條約協定與兩岸協定之內容；立法院的法律全文檢索系統[11]，收藏有法律現行條文資料庫、法律修正過程資料庫與法律條文沿革資料庫；法源法律網[15]，收錄我國全部法學資訊，依資料性質與樣態分為法規、司法判解、行政函令、法學論著索引等四類。

在英美法系的國家，被告有無犯罪係屬事實的認定問題，被告有權要求交由陪審團決定，而台灣法律制

度，則是完全由法官來認定被告的犯罪與否。因此，在台灣一旦有刑事案件發生，經由檢察官的起訴之後，所有的犯罪資料都會經過法官詳細的研讀，然後才由法官作出判決。這樣的做法，可以讓具有專業知識的法官來判定每一個案件，但是法官在處理每一個案件所要花的時間和精神也變多了。

電腦科技的進步帶動了資料的電子化，利用電腦來作文書處理，不但可以節省寫字的時間，更增加了編排的便利性。為方便法官在寫判決書以及查詢資料的方便性，目前台灣已有一套專門設計給法官使用的法官審判作業系統，這套系統每年由司法院負責修訂，並提供最新的法學文件處理功能，但是這套系統的功能，只限於提供各類判決書範本，檢察官的起訴書以及其他相關資料，還是必須由法官親自閱讀整理之後，才能作判決。

每個法官每天都有大量的案件要處理，因此他們也必須花不少時間和精神來研讀每一份資料。倘若機器能夠代替法官研讀起訴書，並給予每個案件建議應適當的法條，進而製作出判決初稿供法官採用，甚至指出檢察官起訴書法條的誤引，相信一定能節省法官研讀資料的時間及精神。本論文便是針對這個目標作了一些研究。我們判決書為訓練資料(training data)找出判斷案由的關鍵詞，以判斷起訴書的案由，再藉由法律條文以及判決書，建立了 rule database，並利用 Longest Common Subsequence[4,9]這一類問題的解法，處理起訴書所記載的犯罪資料與 rule 之間，中文字句語意的比對，最後提出應適用法條的建議。

本文的組織如下。第 2 節為相關研究的回顧，由於國內作學法資訊系統的相關研究不多，因此我們將著重於國外已發展的系統之介紹。在第 3 節中將說明如何以 Longest Common Subsequence 之解法處理中文語意比對的問題。第 4 節將介紹實驗過程與成果。最後在第 5 節我們將作個簡單的結論並討論未來的發展方向。

### 2. 相關研究

在國外，資訊科學與法律結合的應用，已經研究了相當多年的時間[5,10]，如 Risland 與 Ashley 早在 1987 年就建立了專門處理營業秘密法的 HYPO 系統[8]，使用此系統時需要輸入一段事實描述的文字，系統會根據輸入的事實以 case-based reasoning 的方法，找出系統資料庫中相似的案件，並且判斷被告與原告何者為

<sup>†</sup>發表於台灣人工智慧學會第七屆人工智慧與應用研討會論文集第 178 頁到第 183 頁，台灣台中，Nov. 2002。

勝方。另外，Haft, Jones 和 Wetter 也在 1987 年發展了一套法律上的專家系統[2]，這套系統包含了兩大部份：consultation system 和 intelligent tutor，主要的特性是以自然語言的對話方式來達到使用者與機器之間的溝通。系統中的 consultation system 主要用途在於幫助律師了解案件的描述，而 intelligent tutor 的功能是在教導使用者有關法律上的知識。

Haft, Jones 和 Wetter 所發展的系統，主要是架構在其他的系統上，如自然語言的解析器等，其中這套系統在提出時因 consultation system 在 knowledge base 實作上的困難而遇到瓶頸。Rissland 與 Ashley 的 HYPO 系統與我們的研究，主要工作內容都是藉由事實描述來獲得所要的資訊，但 HYPO 系統所處理的是觸犯營業秘密法的案件，這種類型的案件中常有被告與原告之間的爭執的問題，因此 HYPO 系統以 case-based reasoning 的方法找出相似的案件來判斷被告與原告何者為勝訴的一方。而我們的系統所處理的案件皆是由檢察官所起訴，沒有被告與原告之間的爭執，但是卻包含了許多罪型，因此我們著重於犯罪事件應適用法條之建議。

法律文件的解讀牽涉到很廣的層面，理想上一個判決輔助系統應該能夠從事深入的文字分析和解讀。因此牽涉很多語法和語意的分析技術，有許多技術仍然是學者們的研究課題。國內外學者有許多自然語言處理方面的相關研究值得我們引用[例如，1,7]。

### 3. 中文語意的比對

我們的判決製作輔助系統會以起訴書中犯罪事實欄位的資料與 rule database 做比對，若犯罪事實欄位內容所出現的詞彙符合 rule database 中某條 rule 之規定，便可依該 rule 判定應適用的法條。然而中文句子經常出現相差一、兩個字，但語意卻非常近，尤其是在法律的關鍵詞比對上，如：公眾得出入之場所，公眾得出入場所，公眾可出入之場所。在這種情況下，我們無法列舉所有撰寫起訴書的檢察官可能使用的關鍵詞到 rule 中。因此，我們只列出其中最常使用的關鍵詞，在比對事實欄位的內容有無出現 database 中的 rule 的關鍵詞時，再去判斷是否有與關鍵詞語意相近之詞彙出現即可。

要判斷兩句中文句子語意是否相近，就必須找出兩個句子之間的相似度，在此，我們把相似度定義為兩個句子之間依序出現相同文字之最大長度，而這一類的問題即為 Longest Common Subsequence (LCS) 的問題，LCS 的長度愈長，表示兩個句子之間的相似度愈高。在認定起訴書犯罪事實欄位的資料中，是否有與 rule 之關鍵詞語意相近的句子時，我們會先計算起訴書犯罪事實欄位的資料與 rule 之關鍵詞間 LCS 的長度 L，如果

$$\frac{L}{\text{關鍵詞之長度}} \geq H$$

H 為門檻值，我們便認定事實欄位的資料中，曾經出現與關鍵詞語意相近之詞彙。H 的值可以經由練訓資料獲得，在我們的實驗中，經過不斷反覆的測試，發現 H 的值為 0.75 時有不錯的效果。在實作上，我們採用 dynamic programming 的方式實作，下列為我們實作的演算法[3]：

**Input**：給兩個序列  $A=a_1a_2..a_m$  及  $B=b_1b_2..b_n$

**Output**：依序出現相同文字之最大長度

$L(i,j)$  為  $a_1a_2..a_i$  與  $b_1b_2..b_j$  依序出現相同文字之最大長度 ( $0 \leq i \leq m, 0 \leq j \leq n$ )

for i 0 to m do  $L(i,0) = 0$ ;

for j 0 to n do  $L(0,j) = 0$ ;

for i 1 to m do

    for j 1 to n do

        if  $a_i = b_j$  then  $L(i, j) = L(i-1, j-1) + 1$

        else  $L(i, j) = \max\{L(i, j-1), L(i-1, j)\}$ ;

依此演算法，測量「危害他人生命」與「危害他人性命」這兩個句子之 LCS，將得到結果為 5，見表 1。

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
|   | ∅ | 危 | 害 | 他 | 人 | 生 | 命 |
| ∅ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 危 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 害 | 0 | 1 | 2 | 2 | 2 | 2 | 2 |
| 他 | 0 | 1 | 2 | 3 | 3 | 3 | 3 |
| 人 | 0 | 1 | 2 | 3 | 4 | 4 | 4 |
| 性 | 0 | 1 | 2 | 3 | 4 | 4 | 4 |
| 命 | 0 | 1 | 2 | 3 | 4 | 4 | 5 |

表 1 Longest Common Subsequence 實例

### 4. 實驗步驟與結果

為了進行本實驗，我們首先需要搜集判決書以及法律條文，這些資料可以從司法院以及其他與法律相關的網頁上取得。在搜集到需要的資料之後，我們先對資料做前置處理(preprocessing)並找出案由，再依判決書與法律條文建立 rule database，同時從起訴書中找到犯罪事實欄位的資料與被告姓名，並找出起訴書犯罪事實欄位的資料中重要的句子，再從這些重要的句子中，配合 rule database 對該篇判決書之被告下定判決，最後提出應適用法條的建議，實驗流程見下圖 1。

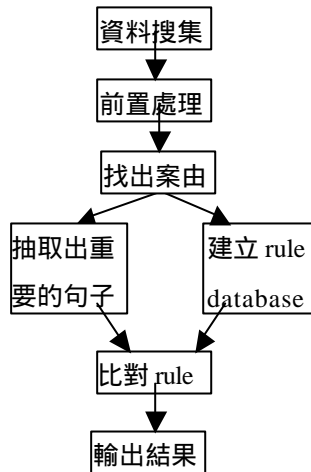


圖 1 實驗流程

#### 4.1 資料的前置處理

從網頁下載回來的資料，通常是無法直接拿來作實驗的。因為網頁的資料在電子化的過程中，可能會因人為的疏失或傳輸的錯誤，造成資料欄位的缺少或者是錯誤。因此我們在取得資料後，必須先對資料作前置處理的工作，主要目的是為了將資料中缺少的欄位補齊，若是資料欄位缺少得太嚴重，例如少了最重要的事實欄位，我們只好捨棄該筆資料不用。

#### 4.2 案由之判斷

在法律規定上，一種類型的罪通常不只有一個法條，例如刑法的賭博罪便包含了刑法第二百六十六條至刑法第二百七十條，對於這些由不同法條所構成的犯罪類型，司法實務上均冠以特定的案由。因為想要機器直接從眾多的法條中去找出適用於某一個案件的法條，那不是很容易的事，所以，我們需要先將判決書當作訓練資料，從判決書犯罪事實欄位的內容，找出辨識案由的關鍵字。我們先從訓練資料中找出每個詞出現的次數，取出現最多次的幾個詞當作關鍵詞，並手動加入部份人工所挑選出之關鍵詞，再以 Longest Common Subsequence 的解法，處理關鍵詞與起訴書犯罪事實欄位的內容之間語意的比對，最後判斷出起訴書所記載被告犯罪事實應予歸類的案由。決定被告犯罪事實的案由可縮小與 rule database 的比對範圍，輔助判決系統便可依被告犯罪事實的案由，從 rule database 中找出相同案由的 rule 來進行比對，進而提供應適用法條的建議。

#### 4.3 建立 rule database

Rule database 是我們判決的標準，因此 rule database 的建立必須從法律條文著手。首先，我們必須從法律條文中取出重要的詞彙，作為 rule 比對的關鍵詞。

要從一份中文的資料中取出關鍵詞，首要的工作，就是對該資料做中文的斷詞處理。我們採取使用中文詞

典的斷詞方法，使用的詞典是 Hownet：首先，將法律條文與中文詞典做比對，凡在詞典上可以查到的，都被當作是一個詞，若有長詞包含短詞的情況，則以長詞為優先，例如「電視廣告」這四個字在 Hownet 中是一個詞，但「電視」與「廣告」分別在 Hownet 中是另外兩個詞，在此我們便認定曾經出現過的詞是「電視廣告」這個長詞，而不是「電視」與「廣告」這兩個短詞。依此方法，我們將法律條文斷成許多中文的詞彙，統計在法律條文中曾經出現過的各個詞彙，分別被幾條條文引用到，再將被引用到之次數高的詞彙刪除，如：意圖、有期徒刑等，刪除這些詞彙的用意在於，若多數法條都有引用到這些詞彙，我們就無法經由這些詞彙的出現來判斷是否適用於某一特定法條。刪除了不適用的詞彙之後，剩下來的詞彙就是我們建立 rule 的基本詞彙。但是只利用法條所引用到的詞彙來當作 rule 的關鍵詞是不夠的，因為一般而言，檢察官在描述事實時所使用的詞彙，不一定會完全跟法條的詞彙相同，有時候會以其他語意上相同的詞彙或句子來表示，為了將這些法條上所沒有引用到的詞彙也抓進 rule 的關鍵詞中，我們再以觸犯同一法條的判決書為訓練資料，訓練出法條沒有引用到的關鍵詞。以上機器挑選出的關鍵詞再加上部份人工挑選的關鍵詞才是我們最後所使用的關鍵詞。有了關鍵詞之後，再來要考慮的是 rule 中的關鍵詞必須被比對到幾個才算符合此 rule，有時候 rule 中的關鍵詞必須全部出現才算，有時候只要出現其中的 50% 就夠了，這個參數的值會因 rule 而不同，可以藉由訓練資料來找出各個 rule 最適宜的參數，調整好這個參數後，再依固定的格式儲存在 database 中，便完成 rule 的建立。

以下讓我們來看一個如何建立 rule 的例子。根據刑法第三百二十條：「意圖為自己或第三人不法之所有，而竊取他人之動產者，為竊盜罪，處五年以下有期徒刑、拘役或五百元以下罰金。意圖為自己或第三人不法之利益，而竊佔他人之不動產者，依前項之規定處斷。前二項之未遂犯罰之。」我們先做中文的斷詞處理，並將之前所提到不適用的詞彙刪除掉，再從中找出適用於此法條之關鍵詞，最後我們可以得到「竊取」這個關鍵詞，另外再加上我們由訓練資料中訓練出來的「竊佔」以及「意圖為自己不法之所有」這兩個關鍵詞，至於符合此 rule 應出現多少關鍵詞之參數的訂定，可以藉由多次訓練找出該參數為何值時，訓練資料會有最好的結果。決定了此 rule 的參數後，所有此 rule 的組成元素都已具備，接下來，我們將建立好的每一條 rule 都存在 rule database 中，為了讓程式可以正確的讀出 rule 所代表的涵意，因此 rule 必須依照固定的格式儲存。在此我們將 rule 的儲存格式作一些規定：第一行必須存放之前所提到的 rule 參數，不管該 rule 有幾個關鍵詞全部存在接下來的幾行中，而最後一行存的是符合該 rule 的法條。依此規定我們將符合刑法第三百二十條的 rule 以下列格式儲存在 rule database 中：

- 1 0.3
- 2 竊取
- 3 竊佔
- 4 意圖為自己不法之所有
- 5 (刑法第三百二十條：意圖為自己或第三人不法之所有，而竊取他人之動產者，為竊盜罪，處五年以下有期徒刑、拘役或五百元以下罰金。)

在此為了方便說明，另外在每行開頭加上行號。第 1 行所存的 0.3 是之前所提的 rule 參數，第 2 至第 4 行是該 rule 的關鍵詞，最後一行也就是第 5 行存的是符合該 rule 的法條。依此儲存格式，程式在讀取 rule database 的資料時，一次完全讀出一條 rule，便可確定該 rule 由多少元素組成，如此可方便得知該 rule 每一行所代表的意義。

#### 4.4 適用法條之建議

要判定一個被告犯了何種罪刑，必須從該被告的犯罪行為來判斷，而犯罪的行為我們可以從起訴書中的犯罪事實欄位來得知。附錄 A 所示為起訴書中犯罪事實欄位內容之例子，為避免個人隱私權之問題，我們將被告姓名以被告 A 表示之。

由於起訴書犯罪事實欄位之內容除了記載被告的犯罪行為外，其他與該案件的相關資料也會一併載入，如此，起訴書犯罪事實欄位的內容將會是很長的一串文字，如果全部拿來與 rule database 中的 rule 一條一條作比對，處理上也會多花一些不必要的時間，為了節省比對 rule 的時間，同時保持判斷結果的正確率，我們必須在起訴書犯罪事實欄位的內容中，抽出重要的句子來處理。首先，將起訴書犯罪事實欄位內容作中文的斷句處理，在此我們斷句的方法是以中文的句號為標準，每當一個句號的出現就是一句話的結束，如此將犯罪事實欄位斷成許多的個別中文句子。然後基於法律案件在描述重要犯罪行為時，一定會標上時間的特性，我們只挑選出有標示時間的句子來進行下一階段的處理。我們判斷一個句子是否有標示時間的方法，是先將該句子依逗號的出現，切成許多小片段，如果在一個小片段之內有依序出現「年、月、日」這三個關鍵字中之任兩者，且關鍵字之間除了「一、二、三、十、同」這幾個字以外，沒有其他字的出現，我們便認為這個小片段為時間的描述，亦即該句子中有時間之標示。在「年、月、日」這三個關鍵字之間，會有「同」字出現是因為當檢察官要描寫兩段時間其年份或月份一樣時，通常第二段被描寫的時間會被寫成「同年」或「同月」。附錄 B 所示為附錄 A 中所挑選出之重要句子。

抽出起訴書犯罪事實欄中重要的句子之後，我們再標示出各句子所描述之被告姓名以及犯罪行為。由於被告的姓名在判決書或起訴書中，都可直接由被告欄位得知，因此我們只要在句子中比對有無該案件之被告姓名出現，就可得知該句子所描述的被告為何者。而犯罪行

為即該句子之全部描述內容。標示結果如附錄 C。加上標示的目的是要方便之後程式的處理。

我們將標示過的句子，依逗號斷成許多小片段，再將各小片段與 database 中的 rule 之關鍵詞比對，比對的過程中以 Longest Common Subsequence 之解法處理中文語意相似度的問題，最後找出適當的法條，結果如附錄 D 所示。

#### 4.5 實驗結果

我們的研究是以簡易刑事案件為主，目前我們採用刑法上「公共危險罪」、「妨害風化罪」、「賭博罪」、「竊盜罪」，以及「毒品危害防制條例」和「兒童及少年性交易防制條例」等 6 種罪型來作判斷案由的實驗。首先，我們各使用了 5 篇同類型案由的判決書來當作訓練資料，從這些訓練資料中，我們分別取得了判定此 6 種類型案由的關鍵詞。在測驗判斷案由的正確率時，我們取了 503 篇起訴書作為測試資料(test data)，其中包含了 153 篇公共危險罪的起訴書，19 篇妨害風化罪的起訴書，25 篇賭博罪的起訴書，94 篇竊盜罪的起訴書，14 篇毒品危害防制條例的起訴書，14 篇兒童及少年性交易防制條例的起訴書，以及 184 篇混合其他類型的起訴書。程式判斷結果之精確率(precision)與召回率(recall)見表 2。

|              | 精確率(precision) | 召回率(recall)  |
|--------------|----------------|--------------|
| 公共危險罪        | 152/156=0.97   | 152/153=0.99 |
| 妨害風化罪        | 18/18=1.00     | 18/19=0.95   |
| 賭博罪          | 20/21=0.95     | 20/25=0.80   |
| 竊盜罪          | 87/90=0.97     | 87/94=0.93   |
| 毒品危害防制條例     | 14/14=1.00     | 14/14=1.00   |
| 兒童及少年性交易防制條例 | 14/14=1.00     | 14/14=1.00   |

表 2 案由判定之精確率與召回率

在適用法條的建議上，我們以刑法第一百八十五條之三(公共危險罪)、第二百三十五條(妨害風化罪)、賭博罪、竊盜罪，毒品危害防制條例第十條，以及兒童及少年性交易防制條例第二十九條來作實驗。判斷適用法條的 rule 之關鍵詞除了由法條中取得，另外各以 5 篇同法條之判決書作為訓練資料，取得法條外的關鍵詞。在測試適用法條之建議的正確率時，我們以前述的 503 篇起訴書為測試資料，先判斷出各篇起訴書的案由，再依照各篇起訴書之案由到 rule database 中尋找判斷同案由的 rule 作比對，以提供應適用之法條的建議。應適用法條之建議的精確率及召回率如下表 3 所示。

|                   | 精確率                  | 召回率                  |
|-------------------|----------------------|----------------------|
| 刑法第一百八十五條之三       | 149/149= <b>1.00</b> | 149/153= <b>0.99</b> |
| 刑法第二百三十五條         | 17/17= <b>1.00</b>   | 17/19= <b>0.89</b>   |
| 賭博罪               | 21/24= <b>0.88</b>   | 21/25= <b>0.84</b>   |
| 竊盜罪               | 94/104= <b>0.90</b>  | 94/94= <b>1.00</b>   |
| 毒品危害防制條例第十條       | 13/13= <b>1.00</b>   | 13/14= <b>0.93</b>   |
| 兒童及少年性交易防制條例第二十九條 | 13/13= <b>1.00</b>   | 13/14= <b>0.93</b>   |

表 3 適用法條建議之正確率

## 5. 討論

本論文利用 Longest Common Subsequence 的解法來處理中文語意比對的問題，並配合 rule database 之建立，進行法律案件的輔助判決。經實驗結果顯示使用 Longest Common Subsequence 的解法可以比對出法律上語意相近的中文句子，並且配合我們的 rule database 做法律案件輔助判決的處理，有相當不錯的效果。

由於資料搜集不易，所以我們目前所做的研究，只能挑選案件較多的罪型來實驗。因此，在未來發展的方面，我們將搜集更多的資料，進行其他罪型的輔助判決之研究。而我們的輔助判決系統是假設一個案件只觸犯一種罪，但是一個案件有時候不只觸犯一種類型的罪，可能有同時犯兩種以上類類型之罪的情形，這種情況在法律上會以併罪或視為裁判上一罪來處理，所以未來我們也將考慮同時觸犯兩種以上類型之罪的處理。在技術上，我們以關鍵詞來建立 rule 進而找出適應用之法條，這樣的作法可以解決大部份的問題，但是觸犯同一法條可能有不同的犯罪行為，如此所佔比例較少的犯罪行為，我們的程式便無法在觸犯同一法條的判決書中找出其關鍵詞，為了解決這個問題，我們也將把各類型的罪犯行為建成立成 case，並且以 case-based reasoning[6]的方法來推論各案件的犯罪行為以判斷出應適用之法條，期望能處理所有類型的犯罪行為並提高正確率。

目前我們的系統只做到法條的建議，對於刑度上的問題仍未處理，原因在於刑度是法官個人主觀的認定，因此，同一個案件若交由不同法官來處理，判刑的輕重也可能會不一樣，所以在刑度判斷的輕重上，我們很難有一個標準可遁。但是未來我們也將朝著比較深層的語意之研究，例如，semantic distance 若能讓機器也可讀懂案件的輕重，那麼在刑度的判決上便可依法條之規定範圍內給予適當的建議。

### 感謝

我們感謝評審對本文所提出的寶貴建議。同時我們也感謝國科會(NSC912213E004013)對本研究的部分補助。

## 參考文獻

- [1] Lee-Feng Chien, "PAT-tree-based keyword extraction for Chinese information retrieval," Proceedings of the 20<sup>th</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50-58, 1997.
- [2] F. Haft, R. P. Jones and Th. Wetter, "A Natural Language Based Legal Expert System for Consultation and Tutoring – The LEX Project," Proceedings of the First International Conference on Artificial Intelligence and Law, pp. 75-83, 1987.
- [3] D. S. Hirschberg, "A Linear Space Algorithm for Computing Maximal Common Subsequence," Communications of the ACM, pp. 341-343, 1975.
- [4] Daniel S. Hirschberg, "Algorithms for the Longest Common Subsequence Problem," Journal of the ACM, pp. 664-675, 1977.
- [5] International Conference on Artificial Intelligence and Law 1987-2001
- [6] J. Kolodner and D. Leake, "A Tutorial Introduction to Case-Based Reasoning," D. Leake, Editor, Case-Based Reasoning: Experiences, Lessons, & Future Directions, pp. 31-65, AAAI Press / The MIT Press, 1996
- [7] M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The Penn Treebank: Annotating Predicate Argument Structure," ARPA Human Language Technology Workshop, pp. 110-115, 1994.
- [8] Edwina L. Rissland and Kevin D. Ashley, "A Case-Based System for Trade Secrets Law," Proceedings of the First International Conference on Artificial Intelligence and Law, pp. 60-66, 1987.
- [9] J. D. Ullman, A. V. Aho, D. S. Hirschberg, "Bounds on the Complexity of the Longest Common Subsequence Problem," Journal of the ACM, pp. 1-12, 1976.
- [10] 陳起行, "德國法資訊學," 法學叢刊, 46(2):79-85, 2001。
- [11] "立法院法律全文檢索系統," <http://lyfw.ly.gov.tw/gaislaw.htm>
- [12] "司法院法學資料全文檢索," <http://wjirs.judicial.gov.tw/jirs/>
- [13] "法務部主管法規資料庫," [http://www.moj.gov.tw/f6\\_frame.htm](http://www.moj.gov.tw/f6_frame.htm)
- [14] "法務部全國法規資料庫", <http://law.moj.gov.tw/>
- [15] "法源法律網," <http://www.lawbank.com.tw/index.php>

#### 附錄 A：起訴書中犯罪事實欄位之內容實例

- 一、被告 A 前曾因違反毒品危害防制條例案件，經臺灣板橋地方法院於民國八十六年四月二十八日，以八十六年度訴字第四三四號判處有期徒刑六月確定，於八十七年一月十六日執行完畢。復因施用第二級毒品安非他命（下稱安非他命）違反毒品危害防制條例案件，經觀察、勒戒後，認無繼續施用毒品之傾向，業經本署檢毒偵字第三九〇九號為不起訴處分確定。惟查被告 A 復基於施用毒品之犯意，於九十年六月五日為警採尿前九十六小時內某時，在不詳地點施用第二級毒品安非他命，於九十年六月五日，經警依「出矯治機構毒品人口調驗工作」採尿而查獲，經依臺灣板橋地方法院九十年毒聲字第四一七一號裁定送臺灣臺北看守所附設勒戒處所觀察、勒戒後，認有繼續施用毒品之傾向。
- 二、案經臺北縣警察局中和分局報告偵辦。

#### 附錄 B：附錄 A 中之重要句子

- 一、被告 A 前曾因違反毒品危害防制條例案件，經臺灣板橋地方法院於民國八十六年四月二十八日，以八十六年度訴字第四三四號判處有期徒刑六月確定，於八十七年一月十六日執行完畢。
- 二、惟查被告 A 復基於施用毒品之犯意，於九十年六月五日為警採尿前九十六小時內某時，在不詳地點施用第二級毒品安非他命，於九十年六月五日，經警依「出矯治機構毒品人口調驗工作」採尿而查獲，經依臺灣板橋地方法院九十年毒聲字第四一七一號裁定送臺灣臺北看守所附設勒戒處所觀察、勒戒後，認有繼續施用毒品之傾向。

#### 附錄 C：標示姓名以及行為

- 一、姓名：被告 A  
行為：被告 A 前曾因違反毒品危害防制條例案件，經臺灣板橋地方法院於民國八十六年四月二十八日，以八十六年度訴字第四三四號判處有期徒刑六月確定，於八十七年一月十六日執行完畢。
- 二、姓名：被告 A  
行為：惟查被告 A 復基於施用毒品之犯意，於九十

年六月五日為警採尿前九十六小時內某時，在不詳地點施用第二級毒品安非他命，於九十年六月五日，經警依「出矯治機構毒品人口調驗工作」採尿而查獲，經依臺灣板橋地方法院九十年毒聲字第四一七一號裁定送臺灣臺北看守所附設勒戒處所觀察、勒戒後，認有繼續施用毒品之傾向。

#### 附錄 D：建議應適用法條

被告 A(毒品危害防制條例 第十條：施用第一級毒品者，處六月以上五年以下有期徒刑。施用第二級毒品者，處三年以下有期徒刑。)