以字型及位置資訊協助獲取 PDF 之論文題目及章節標題

蕭文峰
屏東商業技術學院 資管系
wfhsiao@mail.npic.edu.tw

洪絲敏
屏東商業技術學院 資管系
s99306015@student-mail.npic.edu.tw

摘要

現有擷取 PDF 文件的工具，經常出現文字錯亂之問題。本研究比較了三個免費的 PDF 擷取工具，發現 PDFBox 在文字資訊(含字型與位置)的提供上較完整，雖然其擷取純文字仍然會受 PDF 製作來源不同有所影響。因此本研究透過 PDFBox 由所擷取的字型與位置資訊來協助重組文章，進而擷取題目與章節標題，以供後續的文件探勘利用。由實驗結果發現本研究提出的方法可有效擷取文章之題目及章節標題：期刊題目的 F1-Measure 值為 0.98、研討會為 0.97；期刊之章節標題的 F1-Measure 值為 0.97、研討會為 0.80；且期刊題目的正確率(Accuracy)為 100%，而研討會是 96%。

關鍵詞：題目擷取、標題擷取、PDFBox、字型資訊、位置資訊

以字型及位置資訊協助獲取 PDF 之論文題目及章節標題

壹、前言

PDF(Portable Document Format)是 Adobe Systems 在 1993 年開發的可攜式文件格式，今已成為全球的標準，目前能在任何電腦及手持設備上開啟。PDF 也是目前學術論文最廣泛使用的文件格式。文件探勘主要是從文件中抽取文字來進行，因此能由 PDF 文件中擷取出完整的文字內容是有其迫切的需要。但 PDF 多元的製作來源，使得 PDF 的格式不一，一般的工具在擷取純文字時，經常出現文字錯亂的情形。

一般而言，使用者在處理 PDF 文件時，通常是由閱讀軟體(例如，Adobe Acrobat Reader[1])手動另存新檔來獲取純文字內容，若要處理大量文件時就不適用。此時就需藉助某些套裝軟體或工具來批次處理，在文件探勘領域開發者除了獲取純文字內容外，仍需對內容進行分析(例如，擷取論文題目作者等書目資訊、計算字頻、反文件頻率等)，因此 PDF 擷取的相關應用程式界面(Application Programming Interface, API)套件會更符合此類的需求。目前供 java 程式語言呼叫的純文字擷取工具有 Xpdf[2]、以及 PDFBox[3]兩套工具，且較易取得。本論文的目的即在於選擇一套合適的工具來協助獲取 PDF 論文檔中的純文字內容與標題內容，當成文件探勘的前處理階段。

一、 現有工具的比較

我們首先說明上述這三種工具的優缺點。Adobe Acrobat Reader 可使用另存成文字檔的功能來擷取 PDF 檔案中的文字，它對於擷取部分 PDF 文件有不錯的效果，但經常會取出一行一字的檔案(如圖 1 所示)，雖然擷取順序正確，但使用者還要自行重新整理文章排序，使其還原。Xpdf 對於擷取 PDF 文件雖然有不錯的效果(如圖 2 所示)，但章節標題經常與段落內容連在一起，就閱讀上而言，使用者以原始文章對照可得知章節標題與段落內容的分別，但對於機器而言，卻無法分辨，故與 Adobe Acrobat Reader 一樣，使用者必須手動重新整理文章，才能進行文件探勘。PDFBox 在獲取單欄、雙欄式的 PDF 文件皆有不錯的效果且保留文字資訊，但擷取出來的文章經常出現文字錯亂的情形，舉例來說，擷取出來的文章中標題可能不在使用者所猜測的位置，而是在更之後才出現標題(如圖 3 所示)。

二、 研究目的

PDFBox 在擷取純文字時會保留文字在 PDF 中的字型、位置資訊，且內文會依實際文章的內容次序獲取，而不會受 PDF 配置(例如，雙欄文件)的影響而錯亂。雖然，PDFBox 仍有些不足，過去也有些研究(e.g., Abderrahim AJEDIG et. al. 2011; Pitale & Sharma 2011)認為其它工具更勝於 PDFBox，但本研究認為這些缺點可以透過其擷取出的字型與位置資訊來加以彌補。因此本研究的具體目的有二：一是以 PDF 文件的文字位置資訊來重組文章，二是提出以字型結合位置資訊為法則，以擷取文件的論文題目及章節標題。過

[1] http://get.adobe.com/tw/reader/
[2] http://www.foolabs.com/xpdf/
[3] http://pdfbox.apache.org/

去雖有些學者研究擷取文件標題，但擷取對象為非 PDF 文件(Peng & McCallum 2006; Tokin & Muller 2008)；也有研究提議利用以字型資訊來擷取論文題目(Beel et. al. 2010)；也有學者利用 Hidden Markov Models(HMM)及 Conditional Random Fields(CRF)來擷取論文題目(Hu et. al. 2006)，但本研究則是更進一步地連章節標題一併擷取。

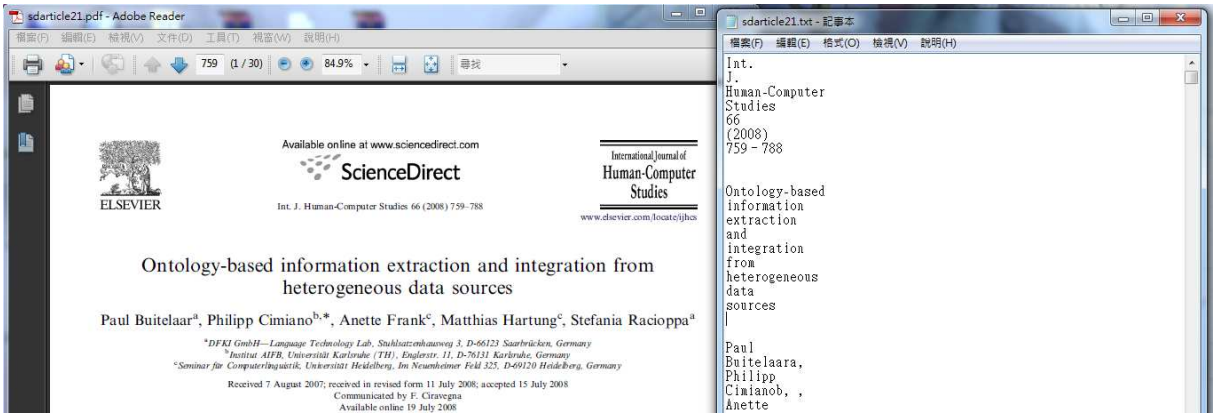

圖 1 Adobe Acrobat Reader 另存文字檔之範例

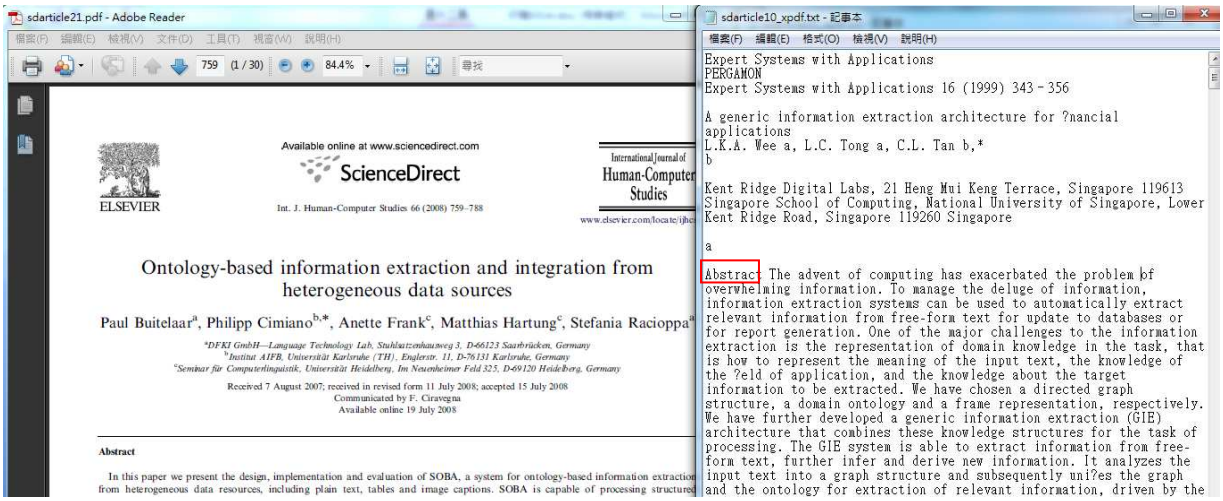

圖 2 Xpdf 擷取 PDF 文件之範例

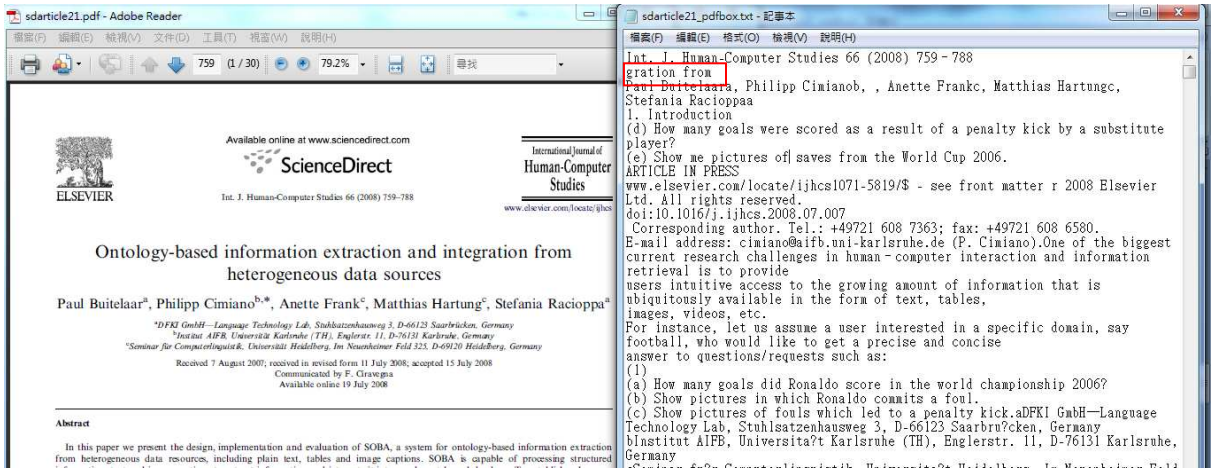

圖 3 PDFBox 擷取 PDF 文件之範例

貳、研究方法

一、 以 PDFBox 提供之文字資訊重組 PDF 文章

PDFBox 提供了相當多關於 PDF 中文字的資訊，但許多資訊是無用的，或是擷取出的資訊是錯誤的，故本研究只利用了 xloc、yloc、font、fontsize、width、height、char 等資訊。char 為文件中的文字，xloc 與 yloc 代表該 char 在 PDF 文件中的 X、Y 座標，font、fontsize、width、height 分別為 char 的字型名稱及大小，寬度以及高度，但 font 所顯示的字型名稱並不等同於我們平常所知的字型名稱(例：Times New Roman、Arial 等)，只可從其中判斷此 char 是否為粗、斜體。我們認為這些資訊可藉由 PDFBox 抽取出來，並且可用於重組文章內容，可大幅度的降低文字的錯亂性，以下是重組文章的法則：

R1. if (word1_yloc == word2_yloc) and (word1_xloc + word1_width == word2_xloc), then word1 與 word2 為相鄰字元

R2. if (word1_yloc == word2_yloc) and (word1_xloc + word1_width != word2_xloc), then word1 與 word2 中間隔了一個空白字元

R3. if (word1_yloc != word2_yloc) and (word1_yloc != word2_yloc), then word1 與 word2 不在同一行中，意即文章到此換行

R4. if (word1_yloc != word2_yloc) and (word1_xloc + 250 < word2_xloc), then word1 與 word2 不在同一欄中，意即文章到此換欄

R5. if (word2_yloc > (word1_yloc + word1.height + word1_height / 2 + word2_height / 2)), then word1 與 word2 之間有兩個換行符號(\n)，屬於不同的段落。

此法則是由於大部分的期刊規定在每個段落間設定 0.5 行的段落間距，或規定標題必須與前段距離為 1 行，因此在此法則下，文章到此會插入兩個\n，表示文章到此段落結束

二、 以字型結合位置資訊為法則之文字擷取法則

由於期刊、研討會都會規定投稿論文格式，並且大致相同，因此以下描述的擷取法則只能適用於一般期刊、研討會論文，若有特殊法則可能就不適用，經研究不同的期刊、研討會規定大致整合如下：

（一） 論文題目應為全文當中最大的字，且位於文章的前十行中。

（二） 1.、2.、1.1 等標號以此類推為章節之標題的起點，且標號後第一個字元為大寫字，或單字為大寫開頭為章節標題的起點。

（三） 章節的標題通常有 Abstract、Introduction、References 等，若單獨出現，或與(二)同時出現，皆視為章節標題。

（四） 論文題目、章節標題等字型有可能為粗、斜體。

（五） 每一層的標題字型資訊應相同。

（六） 新章節通常為一段落的開始，故文章若有換兩行的情形，便有可能出現新的章節標題，此法則是與重組文章法則 R5.相呼應。

參、系統架構與雛型

本系統以 Java 程式開發，並採用 NetBeans (http://www.netbeans.org/) 為開發平台。PDF 文件的文字資訊之擷取是透過 PDFBox，本系統如圖 4 所示是由六個模組所組成，分別描述如下：

File Input：這個模組在系統中負責的是匯入所有的 PDF 文件。

PDFBox Process：這個模組在系統中負責將讀入的 PDF 文件利用 PDFBox 來擷取文字資訊，包括字型及位置資訊。

Record Word Information：這個模組會在系統內部記錄 PDFBox 所擷取出來的文字資訊，也就是在程式內部作為 PDF 文件架構的表示，用於擷取論文題目、章節標題及重組文章。

Extract Model：在此模組，系統會使用擷取法則在 Record Word Information 中擷取標題，在此使用的擷取法則已在研究方法描述過。

Restructure Rule：在此模組，系統會使用重組文章法則在 Record Word Information 中重組文章，在此使用的重組文章法則已在研究方法描述過。

File Output：在此模組中，可將重組完成的文章以及擷取的標題匯出。

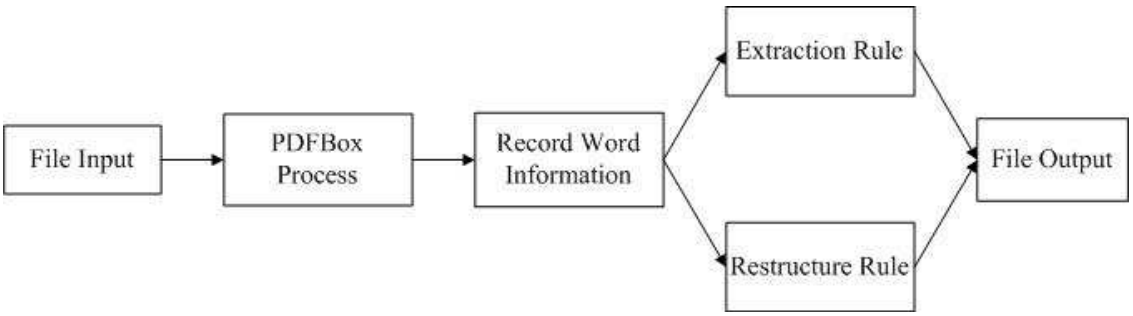

圖 4 系統流程架構圖

本研究所開發之系統雛型(如圖 5 所示)，使用者可利用 File 選單下之 Open File 選項載入 PDF 文件，左邊為已處理完成的檔案列表，右呈現完整文件內容或標題內容，經由選擇標題層數可顯示系統所擷取的文件內容或不同層級的標題內容。若使用者認為系統未擷取出正確標題或想標記重要資訊(例：作者)時，編輯與匯出頁次可讓使用者編輯文件(如圖 6 所示)，利用上方之選單檢視各層級目前的標題內容，方便使用者知道目前系統認為是標題的部分，可以選擇在該行同步插入或刪除標記，也可將編輯完成之文件匯出成 XML 格式，也可利用 File 選單下之 Save Title 選項匯出所有標記為題目及標題的內容。

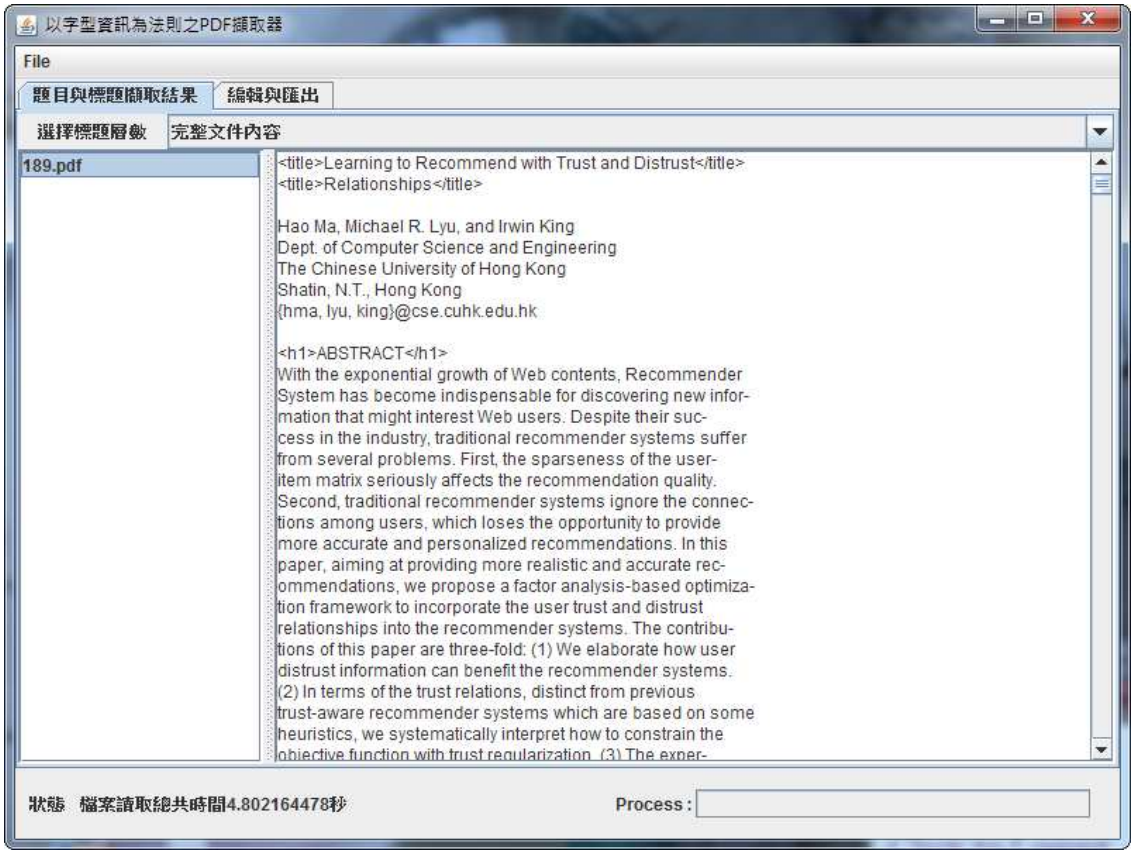

圖 5 系統雛型之題目與標題擷取結果

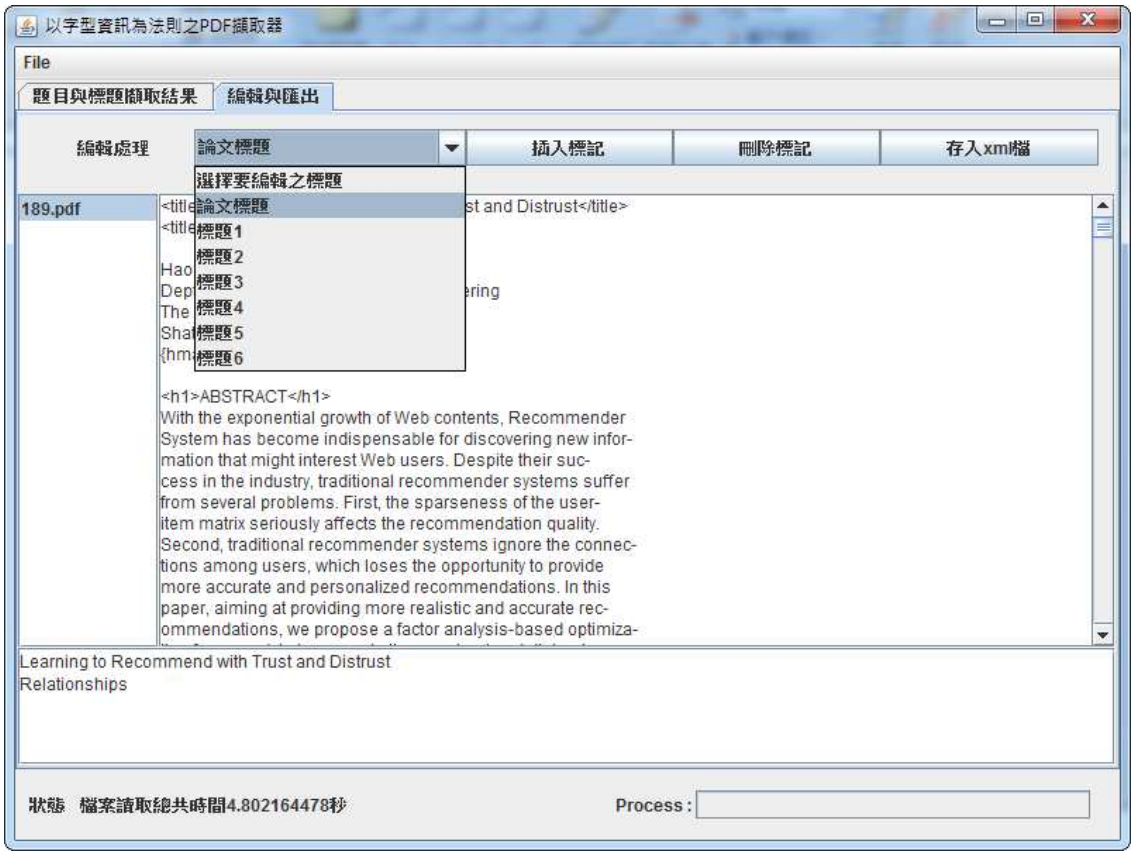

圖 6 系統雛型之編輯與匯出

肆、實驗與討論

一、 重組文章實驗結果與討論

經由圖 7 與圖 3 的對比，可看出本研究所建立的法則可讓文章順序大致上比 PDFBox 原始擷取出的更能按照原始文章順序。但雖然我們以文字的位置資訊為基礎，依照法則來重組文章，但由於是按照 PDFBox 讀入順序來排序的，若要再進一步進行文字探勘必須將註解等不是重要資訊的部分排除，以目前來看，仍然需要更多的法則來支持。

Int. J. Human-Computer Studies 66 (2008) 759–788

Ontology-based information extraction and integration from heterogeneous data sources

Paul Buitelaar
a
, Philipp Cimiano
b,_
, Anette Frank
c
, Matthias Hartung
c
, Stefania Racioppa
a

a
DFKI GmbH—Language Technology Lab, Stuhlsatzenhausweg 3, D-66123 Saarbru?cken, Germany
b
Institut AIFB, Universita?t Karlsruhe (TH), Englerstr. 11, D-76131 Karlsruhe, Germany
c
Seminar fu?r Computerlinguistik, Universita?t Heidelberg, Im Neuenheimer Feld 325, D-69120 Heidelberg, Germany

Received 7 August 2007; received in revised form 11 July 2008; accepted 15 July 2008
Communicated by F. Ciravegna
Available online 19 July 2008

Abstract

In this paper we present the design, implementation and evaluation of SOBA, a system for ontology-based information extraction from heterogeneous data resources, including plain text, tables and image captions. SOBA is capable of processing structured information, text and image captions to extract information and integrate it into a coherent knowledge base. To establish coherence, SOBA interlinks the information extracted from different sources and detects duplicate information. The knowledge base produced by SOBA can then be used to query for information contained in the different sources in an integrated and seamless manner. Overall, this allows for advanced retrieval functionality by which questions can be answered precisely. A further distinguishing feature of the SOBA system is that it straightforwardly integrates deep and shallow natural language processing to increase robustness and accuracy. We discuss the implementation and application of the SOBA system within the SmartWeb multimodal dialog system. In addition, we present a thorough evaluation of the different components of the system. However, an end-to-end evaluation of the whole SmartWeb system is out of the scope of this paper and has been presented elsewhere by the SmartWeb consortium.

圖 7 系統擷取成果

另外，由於 PDF 的內容包羅萬象，在機器學習、財務金融等領域期刊經常介紹許多公式以及公式推導過程，但公式在重組文章上有兩個問題，一是讀入的字元是否等同於原公式的字元，二是原公式的字元是上標或下標，純文字無法表現這樣的格式，故公式的重組目前仍然無法解決。

二、 標題擷取實驗與結果討論

本研究所使用的實驗資料集是 50 篇電子論文，從 ScienceDirect、IEEE Computer Society 等電子資料庫中收集而成，而這 50 篇電子論文其中 25 篇為期刊、25 篇是研討會，藉以研究期刊與研討會不同格式(通常期刊為單欄式文件，而研討會為雙欄式文件)的系統結果。在這兩種文件中各有 5 篇與其它 20 篇的格式不同，藉以比較擷取法則在

特殊格式下的擷取效果。在期刊的 5 篇(EMJ)不同格式的章節標題為大寫開頭，而不是標號開頭；在研討會的 5 篇(ICIC)不同格式的章節標題為 I.、II.、II.等羅馬數字開頭，同樣不是標號開頭。

本研究分別比較論文題目、章節標題及綜合標題擷取結果，並以精確率(Precision)、召回率(Recall)、F1-Measure 來評估擷取效果。(Precision = GoldenFound / AllFound, Recall = GoldenFound / Target, GoldenFound 代表系統所擷取的結果是目標字的字數，AllFound 代表系統所擷取的總字數，Target 代表系統應該擷取的目標字字數，則 F1-Measure = (2 * Precision * Recall) / (Precision + Recall))表 2 與表 3 皆是以擷取法則(一)所擷取的論文題目，可發現大部分皆能正確擷取，但 PDFBox 在擷取 fl、ff、fi 等合體字(Ligature)[4]時，也許是因為製作來源的不同，導致部分能夠成功擷取，部分不能，因此影響了部分文件的擷取結果。此實驗結果也可與 Beel et. al. (2010)的研究比較，該研究認為輕微錯誤(Slight Errors)(例如，fl、ff、fi 等合體字問題)可以不計，但論文題目擷取完全正確才計成正確，若無便不計，故正確率(Accuracy) = 擷取結果完全正確的篇數 / 文件總數。因此，本研究的期刊正確率達 100%，而研討會為 96%(如表 1 所示)較 Beel et. al. (2010)的 77.5%更佳。

表 1 50 篇電子論文題目擷取結果

	Correct		Slight Errors		Total Accuracy	
期刊	21/25	84%	4/25	16%	25/25	100%
研討會	24/25	96%	0/25	0%	24/25	96%

由於期刊與研討會的 Abstract、References 等格式大致不同(舉例來說，Abstract 通常為單獨一行或連接著摘要內容，而 References 有時會是單獨一行；有時與標號一起出現)，因此我們先比較只以擷取法則(二)、(四)、(五)及(六)擷取的章節標題結果。從表 4 與表 5 結果可看出擷取章節標題有不錯的擷取結果，但 ICIC 研討會格式規定圖片及表格的標題也設為粗體，且章節標題非擷取法則所預設的阿拉伯數字為標號，只能以字型資訊做為擷取法則，因此擷取出許多雜訊，擷取效果甚差。ICSE 的擷取結果精確率只有 0.87 左右的原因是在那 5 篇當中有 1 篇不符合 ICSE 的規定，此篇的第一層標題之標號格式錯誤，因此無法利用其與同層標題比對，最終效果不甚理想，拉低 ICSE 整體結果。

最後從表 6 與表 7 可看出加入擷取 Abstract、References 等後導致擷取效果下降，這是因為部分期刊、研討會的 Abstract 此章節標題的表現方式為 Abstract(DSS)或與 ARTICLE INFO 在同一行，系統便會一併擷取；另以 ICIC 此研討會為例，Abstract 此章節標題會與章節內容一起出現，系統在擷取時連同該行章節內容一起擷取，自然效果不佳。因此在綜合比較時，期刊的結果不如只擷取章節標題時好，研討會由於受論文題目擷取效果較佳的影響，反而在綜合效果時有所提升。

[4] fl, ff, fi 皆為兩個字元，但在某些字型下兩個字元會連在一起(故名合體字)，在擷取上會無法判定是什麼字元，所以擷取出的文字會以?呈現。

表 2 各研討會擷取論文題目的平均精確率、召回率、F1-Measure

研討會	精確率	召回率	F1-measure
CCS	1	1	1
HICSS	1	1	1
ICIC	0.81791	0.933333	0.831579
ICSE	1	1	1
RS	1	1	1
平均	0.963582	0.986667	0.966316

表 3 各期刊擷取論文題目的平均精確率、召回率、F1-Measure

期刊	精確率	召回率	F1-measure
DSS	0.978947	0.988889	0.983784
EJOR	1	1	1
EMJ	1	1	1
IPM	0.941013	0.941013	0.941013
JQM	1	1	1
平均	0.983992	0.98598	0.984959

表 4 各研討會擷取章節標題的平均精確率、召回率、F1-Measure

研討會	精確率	召回率	F1-measure
CCS	0.975095	0.987142	0.981022
HICSS	0.9375	1	0.962963
ICIC	0.468097	0.338403	0.361965
ICSE	0.87037	0.780716	0.78107
RS	0.912704	0.953444	0.932421
平均	0.832753	0.811941	0.803888

表 5 各期刊擷取章節標題的平均精確率、召回率、F1-Measure

期刊	精確率	召回率	F1-measure
DSS	0.913087	0.950158	0.928678
EJOR	0.986444	0.990363	0.988384
EMJ	0.968816	0.957095	0.960273
IPM	0.972704	0.976547	0.974606
JQM	0.991667	1	0.995745
平均	0.966144	0.974832	0.969537

表 6 各研討會擷取綜合結果的平均精確率、召回率、F1-Measure

研討會	精確率	召回率	F1-measure
CCS	0.981355	0.990455	0.985851
HICSS	0.948276	1	0.970297
ICIC	0.532519	0.535228	0.497534
ICSE	0.960952	0.817093	0.838013

RS	0.928297	0.952054	0.939689
平均	0.87028	0.858966	0.846277

表 7 各期刊擷取綜合結果的平均精確率、召回率、F1-Measure

期刊	**精確率**	**召回率**	**F1-measure**
DSS	0.751399	0.933446	0.831057
EJOR	0.928433	0.913721	0.920176
EMJ	0.938837	0.96584	0.94742
IPM	0.938837	0.96584	0.94742
JQM	0.992481	0.969556	0.98046
平均	0.902114	0.946175	0.920328

伍、結論

由於現今電子文件(尤其是期刊論文)的標準格式大多是 PDF，在進行文件探勘時必須將 PDF 中的文字內容加以取出才能順利進行。然而，現有工具於擷取 PDF 的純文字上各有優缺點，並未有任何一套免費工具能夠擷取出符合使用者期待的內容。因此，本研究嘗試利用 PDFBox 擷取出來的文字字型與位置資訊，來重組文章內容，進而獲取論文題目及章節標題，以供後續的文件探勘運用(e.g. 可由論文題目及章節標題來決定關鍵詞的重要程度)。

從實驗結果可發現，我們的方法可有效地獲取論文的題目與章節標題，雖然仍有擷取法則不完善(e.g. 未考慮羅馬數字為標號的情況)，及合體字等問題，但整體來說，在擷取期刊論文題目時，正確率幾乎可達 100%(不考慮合體字問題)、研討會論文則可達 96%；在章節標題的擷取上，期刊論文的 F1-Measure 值可達到 0.97、研討會論文則可達 0.80。但在文章重組部分，仍受限於 PDF 格式問題(e.g. 上下標)，還是無法完全將文章不需要的部分去除，這部分仍需手動處理。

未來研究可朝向兩個方向進行，一是利用擷取出來的章節標題及摘要，進行投影片自動產生；二是發展更彈性的法則，或者結合 HMM、CRF 等機器學習法，來協助擷取更完整的標題內容。

陸、參考文獻

1. AJEDIG, A. M., Li, F., Rehman, Au., “A PDF Text Extractor Based on PDF-Renderer,” *Proceedings of the International MultiConference of Engineers and Computer Scientists 2011*, Vol 1, March 2011, pp. 16-18.
2. Beel, J., Gipp, B., Shaker, A., and Friedrich, N., “SciPlore Xtract: Extracting Titles from Scientific PDF Documents by Analyzing Style Information (Font Size),” *Proceedings of the 14th European Conference on Digital Libraries*, volume 6273 of *Lecture Notes of Computer Science(LNCS)*, September 2010, pp. 413-416, (available online at http://www.sciplore.org).

3. Hu, Y., Li, H., Cao, Y., Teng, L., Meyerzon, D., and Zheng, Q., "Automatic Extraction of Titles from General Documents Using Machine Learning," *Information Processing and Management* (42:5), 2006 , pp. 1276-1293.
4. Peng, F., and McCallum., A., "Accurate Information Extraction from Research Papers Using Conditional Random Fields," *Information Processing and Management* (42:4), 2006, pp. 963-979.
5. Pitale, S., Sharma, T., "Information Extraction Tools for Portable Document Format," *International Journal of Computer Technology and Applications* (02:6), 2011, pp. 2047-2051.
6. Tonkin, E., Muller, H., "Keyword and Metadata Extraction from Pre-prints," *Proceedings of the 12th International Conference on Electronic Publishing*, June 2008, pp. 30-44.

Extracting Title and Section Headings from PDF Papers by Utilizing Font and Position Information

Wen-Feng Hsiao
wfhsiao@mail.npic.edu.tw
Department of Information Management,
National Pingtung Institute of Commerce

Ssu-Ming Hung
s99306015@student-mail.npic.edu.tw
Department of Information Management,
National Pingtung Institute of Commerce

Abstract

Current tools for extracting text from pdf documents often suffer from the disorder problem where the text flow is different from the original one especially when the layout is in two columns. In this study we compared three free PDF extraction tools, and found that PDFBox can provide more complete information (in terms of font and position of the text), though its performance might still be influenced by the rendering tools (rendering tools are used to convert document to pdf format). Therefore, we proposed to use the font and position information of text obtained by PDFBox to reconstruct the text order and further to extract the title and section headings. These extracted title and section headings can benefit the subsequent text mining process. The experiment results show that our proposed method can effectively extract the title: The F1-measure of journal papers is 0.98, the F1 of conference papers is 0.97, as well as section headings: The F1-measure of journal papers is 0.97, the F1 of conference papers is 0.80; The accuracy of extracted title for journal papers is 100%, and for conference papers it is 96%.

Keywords: title extraction, heading extraction, PDFBox, font information, position information