Erschienen in: Research and advanced technology for digital libraries :14th European Conference, ECDL 2010, Glasgow, UK, September 6 - 10, 2010; proceedings / Mounia Lalmas et al. (Hrsg.). - Berlin [u.a.] : Springer, 2010. - (Lecture Notes in Computer Science ; 6273). - S. 413-416. - ISBN 978-3-642-15463-8

SciPlore Xtract: Extracting Titles from Scientific PDF Documents by Analyzing Style Information (Font Size)

Jöran Beel[1,2], Bela Gipp[1,2], Ammar Shaker[1], and Nick Friedrich[1]

[1] Otto-von-Guericke University, Computer Science/ITI/VLBA-Lab, Magdeburg, Germany
[2] UC Berkeley, Berkeley, California, USA
{beel,gipp,shaker,friedrich}@sciplore.org

Abstract. Extracting titles from a PDF's full text is an important task in information retrieval to identify PDFs. Existing approaches apply complicated and expensive (in terms of calculating power) machine learning algorithms such as Support Vector Machines and Conditional Random Fields. In this paper we present a simple rule based heuristic, which considers style information (font size) to identify a PDF's title. In a first experiment we show that this heuristic delivers better results (77.9% accuracy) than a support vector machine by CiteSeer (69.4% accuracy) in an 'academic search engine' scenario and better run times (8:19 minutes vs. 57:26 minutes).

Keywords: header extraction, title extraction, style information, document analysis.

1 Introduction

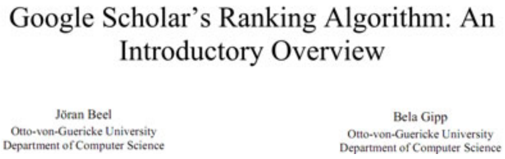
Extracting the title from PDF documents is one of the prerequisites for many tasks in information retrieval. Among others, (academic) search engines need to identify PDF files found on the Web. One possibility to identify a PDF file is extracting the title directly from the PDF's metadata. However, often the PDF metadata is incorrect or missing. Therefore, what is often tried is to extract the title from the PDFs' full text.

Usually, machine learning approaches such as Support Vector Machines (SVM), Hidden Markov Models and Conditional Random Fields are used for extracting titles from a document's full text. According to studies, the existing approaches achieve excellent accuracy, significantly above 90%, sometimes close to 100% [1, 2, 3]. However, all existing approaches for extracting titles from PDF files have two shortcomings. First, they are expensive in terms of runtime. Second, they usually convert PDF files to plain text and lose all style information such as font size.

For our academic search engine SciPlore.org we developed *SciPlore Xtract*, a tool applying rule based heuristics to extract titles from PDF files. In this paper we present this tool, the applied heuristics and an evaluation.

2 SciPlore Xtract

SciPlore Xtract is an open source Java program that is based on pdftohtml[1] and runs on Windows, Linux and MacOS. The basic idea is to identify a title based on the rule that it will be the largest font on the upper first third on the first page.

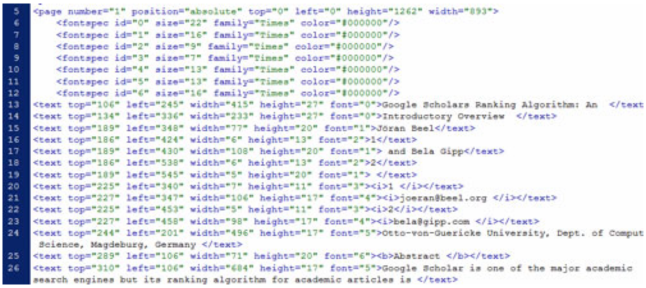


Google Scholar's Ranking Algorithm: An Introductory Overview

Jöran Beel
Otto-von-Guericke University
Department of Computer Science

Bela Gipp
Otto-von-Guericke University
Department of Computer Science

Fig. 1. Example PDF



```
 5 <page number="1" position="absolute" top="0" left="0" height="1262" width="893">
 6     <fontspec id="0" size="22" family="Times" color="#000000"/>
 7     <fontspec id="1" size="16" family="Times" color="#000000"/>
 8     <fontspec id="2" size="9" family="Times" color="#000000"/>
 9     <fontspec id="3" size="7" family="Times" color="#000000"/>
10     <fontspec id="4" size="13" family="Times" color="#000000"/>
11     <fontspec id="5" size="13" family="Times" color="#000000"/>
12     <fontspec id="6" size="16" family="Times" color="#000000"/>
13 <text top="106" left="245" width="415" height="27" font="0">Google Scholars Ranking Algorithm: An  </text
14 <text top="134" left="336" width="233" height="27" font="0">Introductory Overview  </text>
15 <text top="189" left="348" width="77" height="20" font="1">Jöran Beel</text>
16 <text top="186" left="424" width="6" height="13" font="2">1</text>
17 <text top="189" left="430" width="108" height="20" font="1"> and Bela Gipp</text>
18 <text top="186" left="538" width="6" height="13" font="2">2</text>
19 <text top="189" left="545" width="5" height="20" font="1"> </text>
20 <text top="225" left="340" width="7" height="11" font="3"><i>1 </i></text>
21 <text top="227" left="347" width="106" height="17" font="4"><i>joeran@beel.org </i></text>
22 <text top="225" left="453" width="5" height="11" font="3"><i>2</i></text>
23 <text top="227" left="458" width="98" height="17" font="4"><i>bela@gipp.com </i></text>
24 <text top="244" left="201" width="496" height="17" font="5">Otto-von-Guericke University, Dept. of Comput
    Science, Magdeburg, Germany </text>
25 <text top="289" left="106" width="71" height="20" font="6"><b>Abstract </b></text>
26 <text top="310" left="106" width="684" height="17" font="5">Google Scholar is one of the major academic
   search engines but its ranking algorithm for academic articles is </text>
```

Fig. 2. Example XML Output

In the first step, SciPlore Xtract converts the entire PDF to an XML file. In contrast to many other converters, SciPlore Xtract keeps all layout information regarding text size and text position. Figure 2 shows an example XML output file of the PDF showed in Figure 1. Lines 6 to 12 of the XML file show all font sizes that are used in the entire document (in this case it is all "Times" in a size between 7 and 22 points). Below this, each line of the original PDF file is stated including layout information such as the exact position in which the line starts, and which font is used.

SciPlore Xtract now simply needs to identify the largest font type (in the example the font with the ID=0). Which text uses this font type on the first page is then identified and to assumed to be the title.

3 Methodology

In an experiment, titles of 1000 PDF files were extracted with SciPlore Xtract. Then, titles from the same PDFs were extracted with a Support Vector Machine from CiteSeer [1] to compare results. CiteSeer's tool is written in Perl and based on SVM Light[2] which is written in C. As CiteSeer's SVM needs plain text, the PDFs were converted

[1] http://www.pdftohtml.sourceforge.net

[2] http://svmlight.joachims.org/

once with PDFBox[3] and once with pdftotext[4] as these are the tools recommended by CiteSeer. It was then checked for each PDF if the title was correctly extracted by SciPlore Xtract and CiteSeer's SVM (for both the pdftohtml text file and the PDFBox text file). If the title contained slight errors the title was still considered as being identified correctly. 'Slight errors' include wrongly encoded special characters or, for instance, the inclusion of single characters such as '*' at the end of the title.

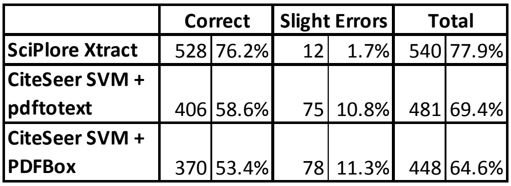
The PDFs analyzed were a random sample from our SciPlore.org database, a scientific (web based) search engine. A title was seen as being correctly extracted when either the main title or both the main title and the sub-title (if existent) were correctly extracted. The analyzed PDFs were not always scientific. It occurred that PDFs represented other kind of documents such as websites or PowerPoint presentations. However, we consider the collection to be realistic for an academic search engine scenario.

4 Results

From 1000 PDFs, 307 could not be processed by SciPlore Xtract. Apparently, SciPlore Xtract (respectively pdftohtml) struggles with PDFs that consist of scanned images on which OCR has been applied. For further analysis only the remaining 693 PDFs were used. We consider this legitimate as the purpose of our experiment was not to evaluate SciPlore Xtract, but the applied rule based heuristic.

For 54 of the 693 PDFs (7.8%), titles could neither be extracted correctly by SciPlore Xtract nor CiteSeer's SVM. Only 160 (23.1%) of the titles were correctly identified by all three approaches. Overall, SciPlore Xtract extracted titles of 540 PDFs correctly (77.9%). CiteSeer's SVM applied to pdftotext identified 481 titles correctly (69.4%). CiteSeer's SVM applied to PDFBox extracted 448 titles correctly (64.6%). Table 1 shows all these results in an overview.

Table 1. Title Extraction of 693 PDFs



	Correct		Slight Errors		Total	
SciPlore Xtract	528	76.2%	12	1.7%	540	77.9%
CiteSeer SVM + pdftotext	406	58.6%	75	10.8%	481	69.4%
CiteSeer SVM + PDFBox	370	53.4%	78	11.3%	448	64.6%

When only completely correct titles are compared, SciPlore Xtract performs even better. It extracted 528 (76.2%) titles completely correct, while CiteSeer's SVM extracted only 406 (58.6%) respectively 370 (53.4%) completely correct.

SciPlore Xtract required 8:19 minutes for extracting the titles. SVM needed 57:26 minutes for extracting the titles from the plain text files (this does not include the time to convert the PDFs to text), which is 6.9 times longer. However, we need to

[3] http://pdfbox.apache.org/

[4] http://www.foolabs.com/xpdf/download.html

emphasize that these numbers are only comparable to a limited extent. CiteSeer's SVM extracts not only the title but also other header data such as the authors and CiteSeer's SVM is written in C and Perl while SciPlore Xtract is written in Java.

5 Discussion and Summary

All three tests show significantly worse results than the often claimed close-to-100% accuracies. Our tests showed (1) that style information such as font size is suitable in many cases to extract titles from PDF files (in our experiment in 77.9%). Surprisingly, our simple rule based heuristic performed better than a support vector machine. However, it could be that with other text to PDF converters, better results may be obtained by the SVM. CiteSeer states to use a commercial tool to convert PDFs to text and recommends PDFBox and pdftotext only as secondary choice. Our tests also showed (2) that runtime of the rule based heuristic was better (8:19 min) than SVM (57:26). However, these numbers are only limitedly comparable due to various reasons.

In next steps, we will analyze why many PDFs could not be converted (30.7%) and in which cases the heuristics could not identify titles correctly. The rule based heuristic also needs to be compared to other approaches such as Conditional Random Fields and Hidden Markov Models. We also intend to take a closer look at the other studies and investigate why they achieve accuracies of around 90%, while in our test the SVM achieved significantly lower accuracies. In the long run, machine learning algorithms probably should be combined with our rule based heuristic. We assume that this will deliver the best results. It also needs to be investigated how different approaches with different languages. Existing machine learning approaches mostly are trained with English documents. It might be that our approach will outperform machine learning approaches even more significantly with non-English documents as style information is language-independent (at least for western languages).

Summarized, despite the issue that many PDFs could not be converted, the rule based heuristic we introduced in this paper, delivers good results in extracting titles from scientific PDFs (77.9% accuracy). Surprisingly, this simple rule based heuristic performs better than a Support Vector Machine based approach.

Our dataset (PDFs, software, results) is available upon request so that other researchers can evaluate our heuristics and do further research.

References

[1] Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., Fox, E.A.: Automatic document metadata extraction using support vector machines. In: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital libraries, pp. 37–48 (2003)

[2] Hu, Y., Li, H., Cao, Y., Teng, L., Meyerzon, D., Zheng, Q.: Automatic extraction of titles from general documents using machine learning. Information Processing and Management 42(5), 1276–1293 (2006)

[3] Peng, F., McCallum, A.: Accurate Information extraction from research papers using conditional random fields. Information Processing and Management 42(4), 963–979 (2006)