Enhanced Browsing System for Electronic Theses and Dissertations

Venkat Srinivasan
Virginia Tech, Blacksburg, VA 24061 USA
svenkat@vt.edu

Mohamed Magdy
Virginia Tech, Blacksburg, VA 24061 USA
mmagdy@vt.edu

Edward A. Fox
Virginia Tech, Blacksburg, VA 24061 USA
fox@vt.edu

ABSTRACT

Electronic Theses and Dissertations (ETDs) can be a valuable aid to learning and scholarship. However, current systems that provide access to ETDs only provide a full text and/or metadata based search and browse facility, thereby limiting ways in which users can interact with and make use of such collections.

Long documents like ETDs can be viewed as containing various streams of information - textual content (in chapters, table of contents, etc.), tables, images, references, etc. In addition to presenting the ETD as a whole to the user, we present the various streams of information in a synergistic fashion, in order to enhance browsing and comprehension.

We describe a design and a prototype for an enhanced web based ETD browsing system, called ETD-Enhance. It allows users to browse and interact with various streams of information present in ETDs, in an integrated fashion. To allow this, we have developed tools to extract individual chapters, figures, captions, etc. from these PDF files. For our pilot study, we have prototyped ETD-Enhance on a small collection of ETDs, and have made the prototype available for public viewing and to collect feedback. We plan to research better techniques for extracting different information streams, as well as better support for user interaction, and to extend our work to the entire collection of ETDs.

Keywords (Required)

Electronic theses and dissertations, text extraction, browsing, exploration, content streams.

1. INTRODUCTION

The Networked Digital Library of Theses and Dissertations (NDLTD) [1] has provided easy access to a growing collection of ETDs. Currently, Scirus[2] and VLTS[3] interfaces exist for users to be able to access this collection. Scirus allows keyword searches only, of metadata and/or full-text. VTLS allows keyword searches, in addition to faceted browsing, both using the metadata records (Figure 1).

Utilization of this collection can be vastly increased by providing better and more user friendly access. This has been a goal of recent research at Digital Library Research Laboratory at Virignia Tech. As part of this initiative, in this paper we propose a design for a tool called ETD-Enhance that is aimed at making reading and comprehension of ETDs easier.

Accordingly, we extract various information streams – viz. text, images, references etc. from ETDs. The various streams of information are presented to the users using a web demo to make the reading and comprehension of ETDs easier.

In the rest of this section we give an outline of the various processes involved in the development of ETD-Enhance.

1.1 Extracting Information Streams from ETDs

ETDs are generally submitted in (or converted eventually to) PDF format. So the first step in our process is to develop tools to extract various information streams that we are interested in, from PDF files. PDF documents however are encoded in the PDF ISO standard, and extracting these streams of information can be quite challenging. Using a combination of several open

source tools, we have successfully developed a methodology to extract individual chapters, images, image captions, references, etc.

1.2 Interactive Document Browsing

As mentioned above, under the current system, users can either search the full text and/or metadata or browse by metadata. The ETD by itself is presented as one single file for linear viewing. It can sometimes be challenging to read and comprehend a ~100 page document when presented in such a fashion. In order to overcome these limitations, we have developed a web demo which presents various components of the ETDs for viewing and user interaction.

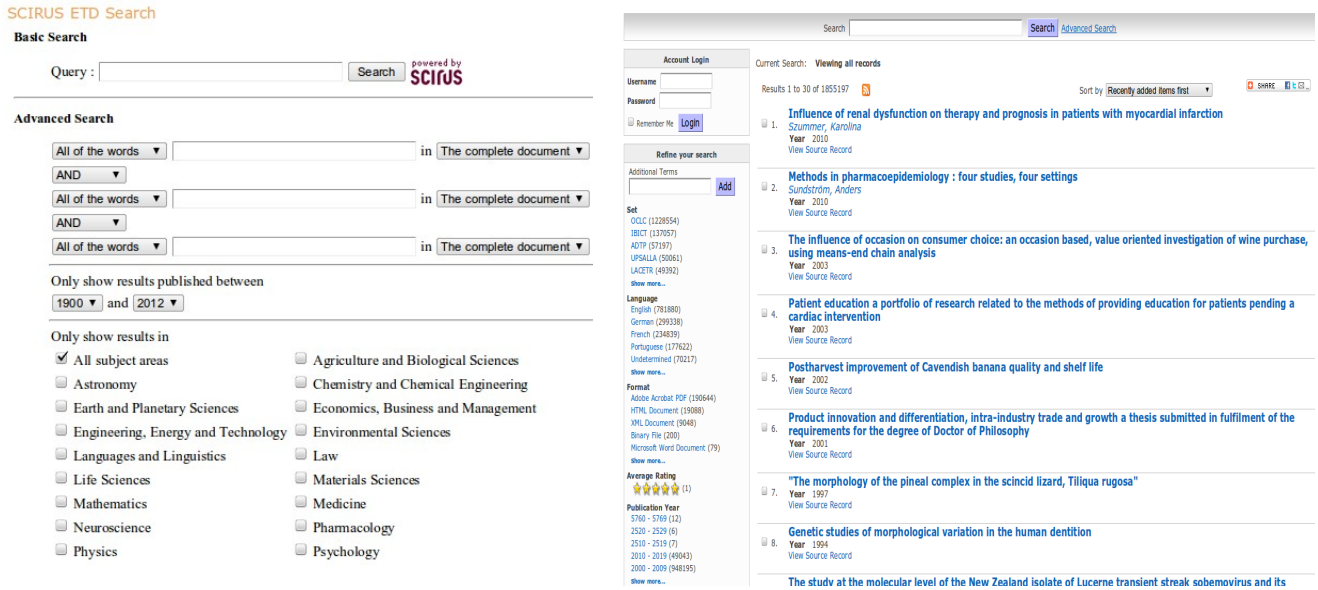

Figure 1. Scirus and VTLS ETD Search interfaces

2. RELATED WORK

2.1 Document Segmentation

The document segmentation problem has received a fair amount of attention [4,5]. The typical approach involves approximate string matching, e.g., the text occurring in table of contents of a document is matched against the text occurring in the body of the document in order to identify candidate chapter boundaries.

These approaches, however, typically either have low accuracy and/or are not generally applicable to other domains or data sets. The sources of error include incorrect identification of table of content entries, wrong mapping between physical and logical page numbers, etc.

A major limitation of these approaches is that they do not make use of font related information. Font characteristics like size, case, location etc. can be quite valuable when segmenting the PDF documents, and sometimes can directly give away chapter boundaries, thereby increasing the accuracy of the method.

Our proposed approach involves combining font related information along with semantic information occurring in the document, like possible chapter headings ("Introduction", "Summary", etc.) occurring in ETDs, in order to segment ETDs. To the best of our knowledge, this is the first attempt at leveraging these disparate sources of information in order to implement document segmentation on ETDs.

2.2 Interactive Document Browsing

The journal Cell recently released a prototype for interactively viewing research papers (Figure 2) [6]. Instead of presenting the whole paper as a single file for linear viewing, various sections that typically occur in a research paper, like background work, summary, results, etc. are presented in separate tabs. This, and many other features like an image gallery, hyperlinked references, etc., provide special support for those working with research papers. This effort from Cell is one motivation for

the research presented in this paper.

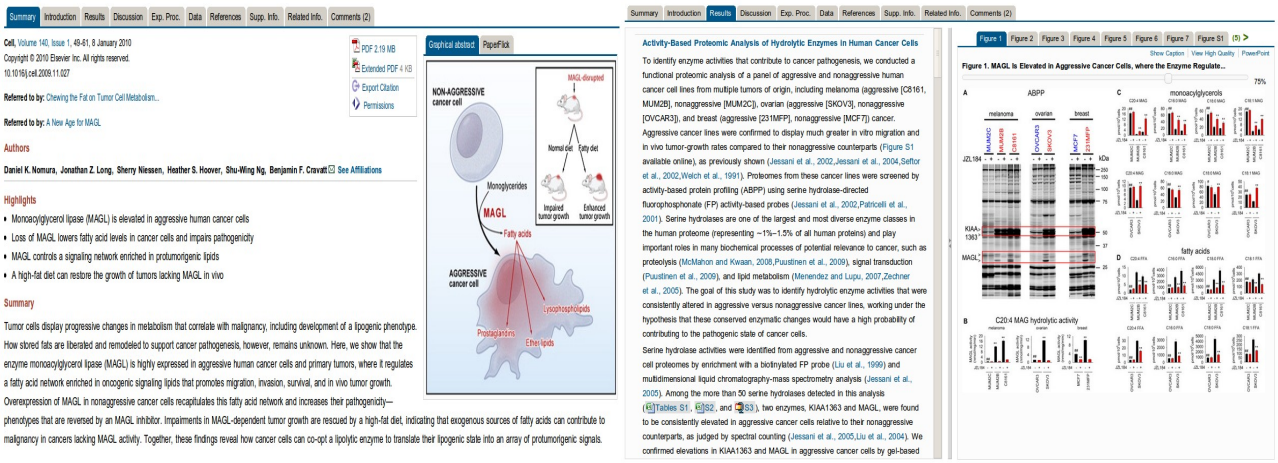

Figure 2. A Recent Prototype available at Journal Cell's website

3. METHODS

We have used several open source tools in order to segment documents and extract images (Table 1). In the following sections, we discuss our methodology in detail. The various steps are shown in Figure 3.

Tool	Source	Function
pdf2xml [7]	SourceForge	Extract font related metadata
pdfimages	XPDF (Linux)	Extracts images (in PPM PBM format) from PDF files
pnmtojpeg	Linux utility	Converts PPM and PBM images into jpeg

Table 1. Open source software used

3.1 Extracting Chapters

In order to extract text from ETDs, we used pdf2xml. It decodes the structure of PDF documents, and produces very fine grained font related information (metadata) about every token (word) that occurs in a PDF file. Sample metadata produced for an example token in a PDF file is shown in Figure 4.

As can be seen, this output is not readily usable. Instead of producing lines or paragraphs that occur in PDF files as output, it generates tagged tokens (words). A page in a PDF file is treated as a co-ordinate space, with the origin (0,0) being at the top left corner of the page. Every token is encoded as a point in XY space, relative to the origin (as seen in "x","y" in Figure 4).

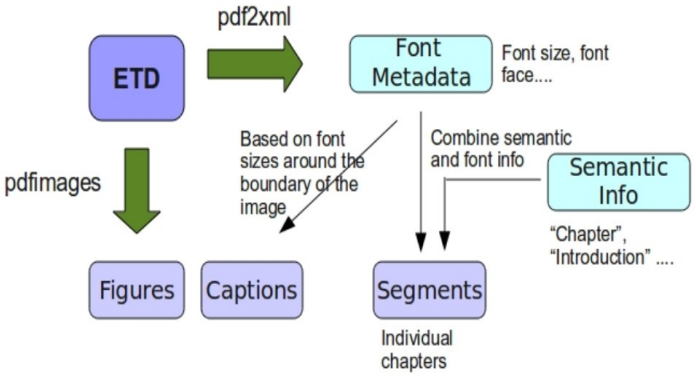

Figure 3. Various steps in text and image extraction

In order to identify chapter boundaries, we first wrote parsers to decode the pdf2xml output and to identify Title Case (TC)

```
<TOKEN sid="p1_s8" id="p1_w4" font-name="liberationserif" symbolic="yes" bold="no" italic="yes" font-size="9" font-color="#000000"
rotation="0" angle="0" x="121.843" y="36.381" base="44.4" width="19.485" height="9.963">name</TOKEN>
```

Figure 4. XML metadata for the token "name" occurring in a PDF file

tokens. We define TC tokens as those tokens that are likely to be chapter titles. ETDs, though, commonly contain fonts with heterogeneous sizes and characteristics. Hence, identifying TC tokens is not straightforward. In order to do this, we detect candidate TC tokens that occur near the top of a page, and use these to identify chapter boundaries. Often, ETDs also have one or more words like "Introduction", "Summary", "Chapter", "References", etc. in chapter headings. We use this cue also to identify TC tokens.

3. 2 Extracting Images and Captions

In order to extract images from PDF files, we make use of pdfimages, which comes bundled in Linux, as part of the XPDF package. We process the PDF file page by page, and extract images that occur in each page. We are interested in extracting image captions also. Hence, once a page is found to contain image(s), we extract the XML metadata information for the page using pdf2xml. Using this metadata information, we identify the XY location of the image(s) on the page and identify small sized texts in the neighborhood of the images as possible image captions.

The images extracted by pdfimages, however, occur in ppm, pnm or vec formats. In order to convert them to jpeg (for ease of display on the web), we use pnmtojpeg.

3.3 Extracting Other Miscellaneous Information

As an add on, we also attempt to extract individual entries in the bibliography section and their corresponding references within the body of the ETD. We identified 3 major styles used in ETDs (Table 3), and wrote parsers to scan through the body of the ETD to identify locations of citations to references.

Style	Example
Bracketed	[Fox2011]
Numeric	[1]
Bracketed	(Fox2011)

Table 3. Major reference styles used in ETDs

3.4 Web Prototype Design

We developed our web prototype using the content management system Drupal [8]. We used several Drupal modules, like those for taxonomy, image gallery, etc. (Table 4) in order to achieve the desired functionality. The users can browse by chapter (or pages), browse the figures, references, etc. (see Section 4.1).

Module Name	Function
Views	Image Gallery
Taxonomy	Taxonomy for browsing by chapter
Vocabulary	Creating navigation block for Taxonomy

Table 4. Drupal modules

4. RESULTS

4.1 Web Demo

Our web demo allows for browsing by separate streams (chapters, images, etc.), as well as presents a unified view of the entire document. The screenshots below (Figure 5) show the use of taxonomy and image gallery features. The web demo can be accessed at http://zappa.dlib.vt.edu/etd/ .

4.2 Evaluation

One critical issue in the development of such a system, besides its usability, is the performance of the backend methods. In order to understand how accurate our text, image, and caption extraction methods are, we ran several experiments.

To evaluate the accuracy of the document segmentation technique, we randomly selected 10 ETDs each from the Engineering, Arts, Business, and Mathematics disciplines from the Virginia Tech ETD collection. Our algorithm achieved an accuracy of 70%, 50%, 70%, and 60% respectively on this data set. We consider the algorithm to be accurate when it successfully identifies every single chapter boundary for an ETD – so for example in this case, our algorithm perfectly identified every single chapter boundary in 7 out of the 10 Engineering ETDs we had selected. If we relax the criterion a little bit, and allow for identification of some but not all chapters in ETDs, the accuracy goes higher, although more experiments are needed to get a good estimate. Nevertheless, out of 40 ETDs used in the experiment, our algorithm perfectly segmented 25 of them.

Evaluating the performance of our image and captions extraction tool is a little harder. One problem is that pdfimages extracts certain extra ghost images from ETDs. These are just small sized spurious ppm or vec files, and without visual inspection, it is hard to tell whether it is a real image extracted from the ETD or just some ghost image. We observed, however, that in the case of ETDs in our collection, the ghost images are mostly <1KB in size. So, in order to perform a reasonable evaluation of our method, we ignore all extracted image files that are less than 1KB. Another problem is that pdfimages segments the images themselves under certain circumstances. For example, when the image is a flowchart, pdfimages extracts certain segments separately, and returns multiple images instead of the entire flowchart as a whole.

To get a rough estimate of the performance of our image and caption extraction tool, we selected 10 ETDs at random from the Virginia Tech ETD collection, and extracted images (and captions) from them. These ETDs were found to contain a total of 91 images, out of which we were able to recover 36 images. The rest of the images either could not be recovered at all, or were recovered only partially or were segmented. Of these 36 images, we were able to extract captions for 24 of them. More experimentation (including user studies) is needed to get better estimates.

Figure 5. Web demo

5. CONCLUSIONS AND FUTURE WORK

In this paper we present the design and a prototype of an enhanced document browsing system for ETDs. Using open source software, we developed tools to extract individual chapters, images, and other auxiliary information like individual references, image captions, etc. While our tools achieve good performance in general, under certain circumstances they do not. In order to improve on this, we are looking at using commercial PDF manipulation software, TET in particular, which is known to have good performance especially with image extraction.

One other important area of future research is to perform user studies on the web demo in order to better understand the effects on typical users of ETDs.

REFERENCES

1. Networked Digital Library of Theses and Dissertations (NDLTD), http://www.ndltd.org/ , Retrieved June 2011

2. Scirus ETD Search, http://www.ndltd.org/serviceproviders/scirus-etd-search, Retrieved June 2011

3. VTLS Visualizer, http://thumper.vtls.com:6090/, Retrieved June 2011

4. Dejean H. and Meunier J.L. 2005. Structuring documents according to their table of contents. In Proceedings of the 2005 ACM symposium on Document engineering (DocEng '05). ACM, New York, NY, USA, 2-9.

5. Lin, X. and Xiong, Y. 2006. Detection and analysis of table of contents based on content association. In Proceedings of IJDAR. 132-143.

6. Cell: Article of the Future , http://www.cell.com/abstract/S0092-8674(09)01439-1#Summary , Retrieved June 2011

7. pdf2xml Converter, http://sourceforge.net/projects/pdf2xml/ , Retrieved June 2011

8. Drupal, http://drupal.org , Retrieved June 2011