Meta-Content framework for back index generation

Tripti Sharma, Assistant Professor
Department of computer science
Chhatrapati Shivaji Institute of Technology.
Durg, India
triptisharma@csitdurg.in

Sarang Pitale, Assistant Professor
Department of Information Technology
Bhilai Institute of Technology.
Durg, India
mail@dsarang.com

***Abstract*— Book reading is a common thing which every one of us does in our life. A common strategy to spot a page for reading is to use front index and back index. A front index generally contains the sections and subsections topic with their corresponding page numbers. A back index contains various words of books with currosponding page numbers in the sorted alphabetical order. From back index and front index page numbers are identified for topic spotting. The proposed paper presents a Meta content framework for for generating back indexes for e books which uses part of speech tagging.**

Keywords-front index, back index, topic spoting, part of speech tagging

I. Introduction

Back index generation is one of the useful tasks in book writing; it provides a quick reference to a query word. It is observed that most of the words in back index are comprised of nouns, with page numbers where they occurred. Meta-content framework uses the same technique. Technique extracts all the nouns present in a page combine with the page number and repeat this same operation for all the pages. The technique is applied to e-books. E-books are the digital version of books available in physical form. Proposed technique uses e-books in portable document format because it is one of the favorite's formats of information exchange on internet. PDF [1] has various advantages over other formats .Security and object wrapping are some of such advantages. But these advantages are the road blockers for back index generation process. This advantage restricts the information extraction process. Present scheme uses iText [2] library for extracting information from PDF format e-books. After conversion Stanford Part-Of-Speech Tagger[3] is used for extracting nouns from the text. Meta content framework streamlines all these processes in such a way that it generates a simple back index for an e-book. Working of Meta content framework for back index generation is shown in form of flow diagram in figure 1.

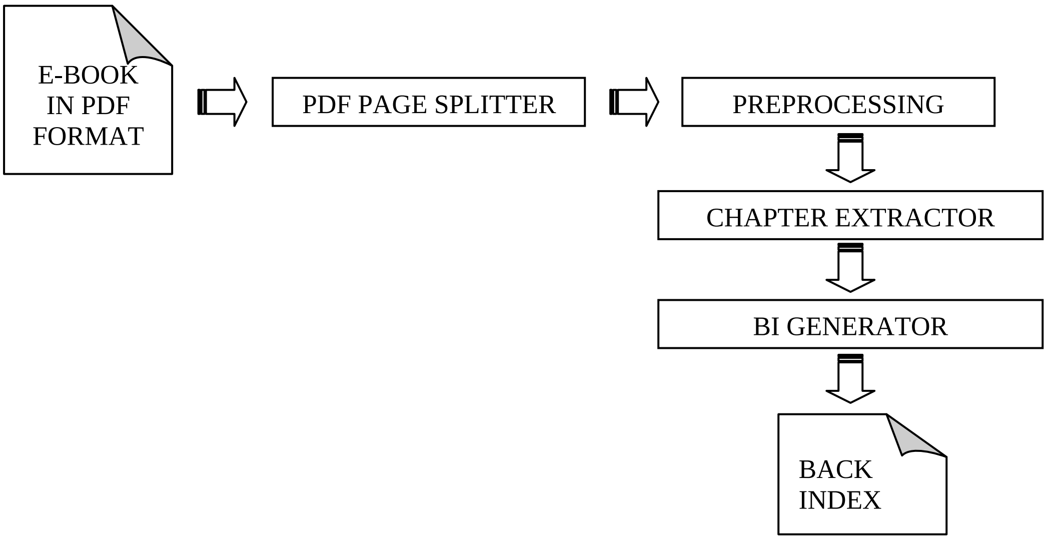


Figure 1. Meta Content Framework for Back Index Generation

II. USER INTERFACE

The Meta content frame work is developed using Java Technology [4]. The master user interface is shown in figure 2.The master user interface comprises of various menus which are listed below:

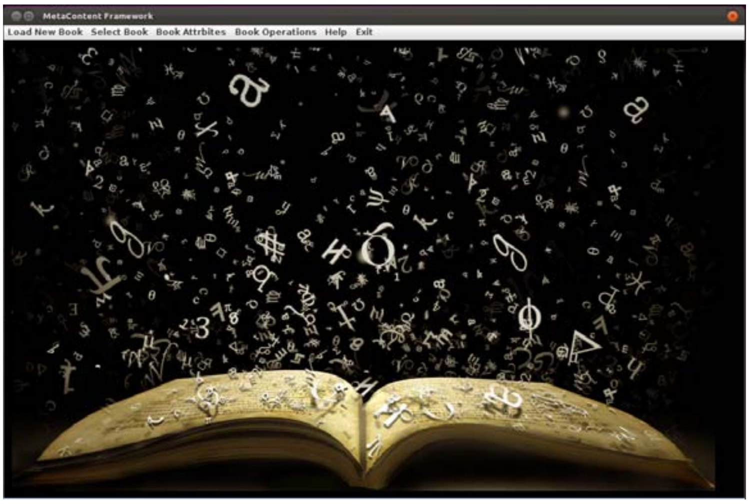
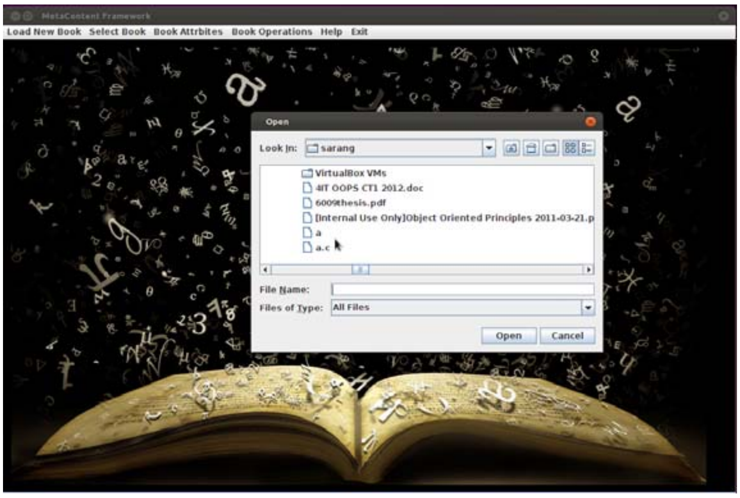


Figure 2. Master user interface of Meta Content framework

A. Load Book:

This operation is used to load a PDF e-book to the frame work. After selecting this option a file-chooser helps the user to select a book from the drives, as shown in figure 3.



Look In: sarang

VirtualBox VMs
4IT OOPS CT1 2012.doc
6009thesis.pdf
[Internal Use Only]Object Oriented Principles 2011-03-21.p
a
a.c

File Name:

Files of Type: All Files

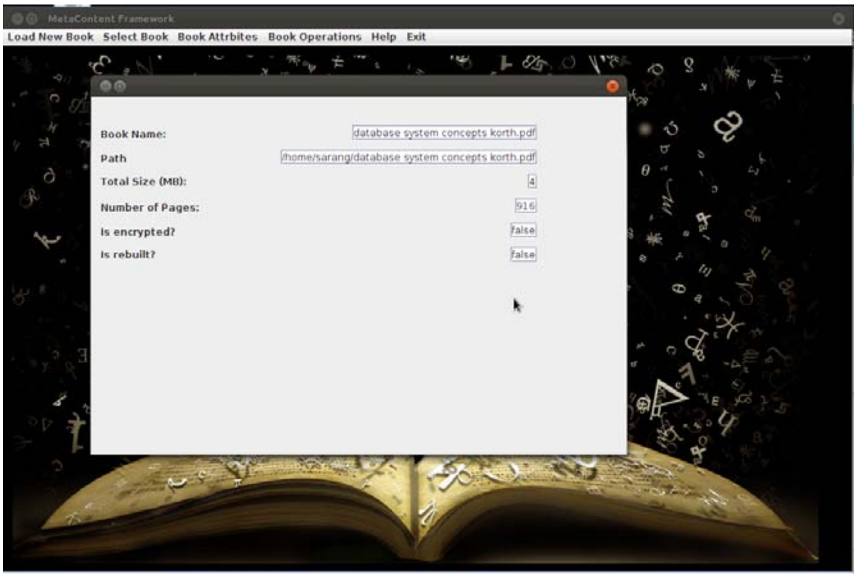
Figure 3. Loading file in framework

B. View attributes:

This operation lets user to view different attributes of the selected e-book. Various attributes of an e-book are:

1) *Name*
2) *Size*
3) *Number of pages*
4) *Encryption status*
5) *Rebuild status*

The user interface to view attributes of an e-book is shown in figure 4.



Book Name: database system concepts korth.pdf
Path /home/sarang/database system concepts korth.pdf
Total Size (MB): 4
Number of Pages: 916
Is encrypted? false
Is rebuilt? false

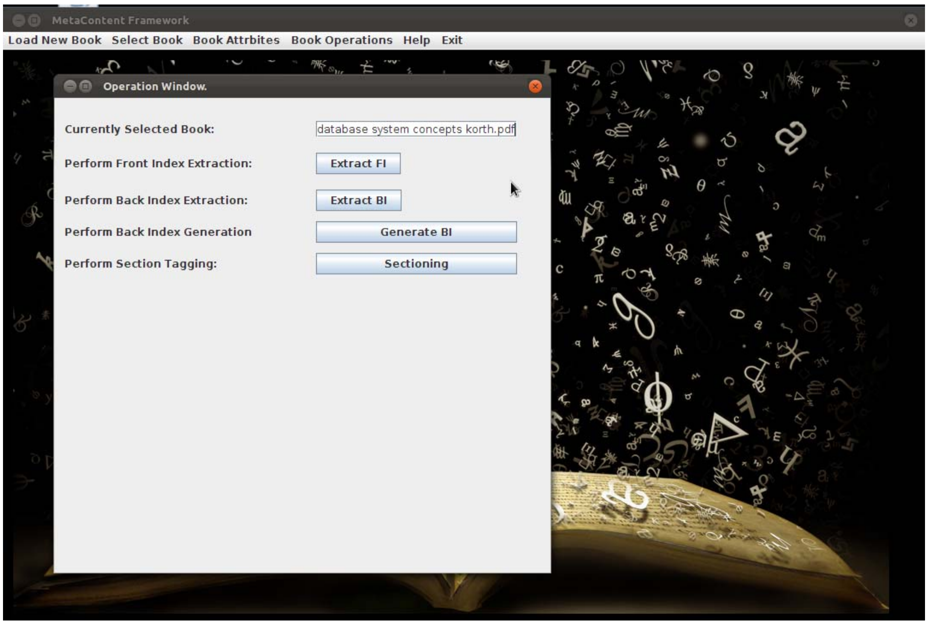
Figure 4. Attribute vision using Meta Content Framework

C. *Book operations:*

This option lets the user to perform various extraction and generation based operations on e-books . such as :

1) *Front Index Extraction*
2) *Back Index Extraction*
3) *Back index Generation*
4) *Sectioning*

The interface for this option is shown in figure 5.



Currently Selected Book: database system concepts korth.pdf

Figure 5. Book operations using Meta Content Framework

D. *Help and Exit:*

These are the general options which are same as the other available software’s. This option includes:

1) *About: Shows the general detail about the framework*

III. INPUT FILE FORMAT SELECTION

Selection of input file format is a crucial step. Selection of file format for e-books should be in such a way that the format should be platform independent, open standard, digital rights management support, image and table support. All these features are only supported by PDF i.e. portable document format. Hence the proposed framework uses PDF as an input format for e-books.

IV. FILE SPLITTER

This module splits the e-book of PDF format into number of pages which the book contains. For the splitting purpose a java based library iText is used which splits the PDF file into number of pages it contains. The process begins with the creation of a directory named exactly the same as of the file. After creating the directory the Meta Content framework splits the input file into number of pages and saves them into the directory. The naming convention of the book is as “book_name-Page-No.-pdf”.

Finally, after completing the split process the inputs were ready for the next process. The split process is important because it provides an unique identification of the page and its respective contents.

V. PREPROCESSING

After the successful split process all the PDF pages are preprocessed for the conversion process. The preprocessing step is combination of three steps:

1) *Image Segmentation*
2) *File management*
3) *Text Conversion*

Image Segmentation is one of the crucial steps in Meta Content framework. This step detects the co-ordinate of an image in a PDF page. After getting the coordinates the schemes calculates its height and width, creates an html tag and write the tag in PDF file with the same name.

In between the segmentation process the scheme continuously deletes the old file and writes the new file until al the images had segmented. This task is done by File management step.

Text conversion is the last step of preprocessing. Various PDF to Text extraction tools [5] are available which have certain advantages and disadvantages. In this step a java based library, iText, is used for the extraction of text contents from the PDF files. iText is used by César García-Osorio et.al.[6] For developing A Tool for Teaching LL and LR Parsing Algorithms .

Preprocessing is the most important part of the Meta content framework, because the output of the preprocessing module is acts as an input for the rest of the modules framework. If the desired output is not generated from the preprocessing module, rests of the modules were fails to operate.

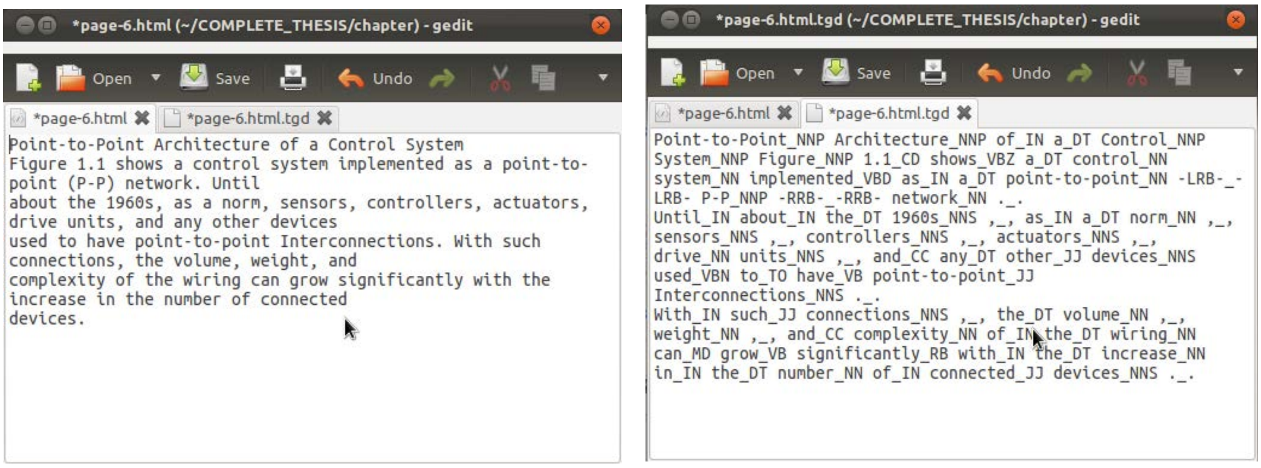
VI. CHAPTER EXTRACTOR

Output text files generated by the preprocessing module are now traversed to select those pages which are contributed as main chapters. This module extracts those pages which are the major contributing factors for back index generation. It is observed that the back index is mainly comprises of nouns from the chapters not from the references, acknowledgement etc. hence the module uses the same fact and separate those pages which are the part of chapters.

To perform such a task Meta content frame work uses a supervised String matching algorithm, which generally searches those pages which indicates the start and end of the chapter contents. After the pages were found, they are separated and saved in a directory named as chapter.

VII. BACK INDEX GENERATOR

This module is the final module for back index generation. The chapter content generated by the previous modules takes the part in back index generation. As discussed previously the back index mainly contains nouns, the major part of the module is to fetch the nouns. To fetch the nouns, Stanford Part Of Speech tagger [7] is used. Part of speech tagger shows the each word or chunks with part of speech. A simple output generated by the Stanford part of speech tagger is shown in figure 6. It is observed that the nouns of the text file are combined with “_NNP”; this can be observed in figure 6. The text file is one by one processed with Stanford tagger, tagger generates file with .tgd (i.e. tagged) extension. Now after .tgd file generation scheme extracts the word chunks with combination “_NNP”. Each of the extracted chunks is combines with the page number and writes them in a file named as “Filename_BI”. At last the back index file is sorted in alphabetical order to form the final sorted back index.


*page-6.html (~/COMPLETE_THESIS/chapter) - gedit
Open
Save
Undo
*page-6.html
*page-6.html.tgd
*page-6.html.tgd (~/COMPLETE_THESIS/chapter) - gedit

```
Point-to-Point Architecture of a Control System
Figure 1.1 shows a control system implemented as a point-to-
point (P-P) network. Until
about the 1960s, as a norm, sensors, controllers, actuators,
drive units, and any other devices
used to have point-to-point Interconnections. With such
connections, the volume, weight, and
complexity of the wiring can grow significantly with the
increase in the number of connected
devices.
```

```
Point-to-Point_NNP Architecture_NNP of_IN a_DT Control_NNP
System_NNP Figure_NNP 1.1_CD shows_VBZ a_DT control_NN
system_NN implemented_VBD as_IN a_DT point-to-point_NN -LRB-_-
LRB- P-P_NNP -RRB-_-RRB- network_NN ._.
Until_IN about_IN the_DT 1960s_NNS ,_, as_IN a_DT norm_NN ,_,
sensors_NNS ,_, controllers_NNS ,_, actuators_NNS ,_,
drive_NN units_NNS ,_, and_CC any_DT other_JJ devices_NNS
used_VBN to_TO have_VB point-to-point_JJ
Interconnections_NNS ._.
With_IN such_JJ connections_NNS ,_, the_DT volume_NN ,_,
weight_NN ,_, and_CC complexity_NN of_IN the_DT wiring_NN
can_MD grow_VB significantly_RB with_IN the_DT increase_NN
in_IN the_DT number_NN of_IN connected_JJ devices_NNS ._.
```

Figure 6. Figure 6 Part of speech tagging of a sample text.

The working of back index generation can be observed by Figure 7.

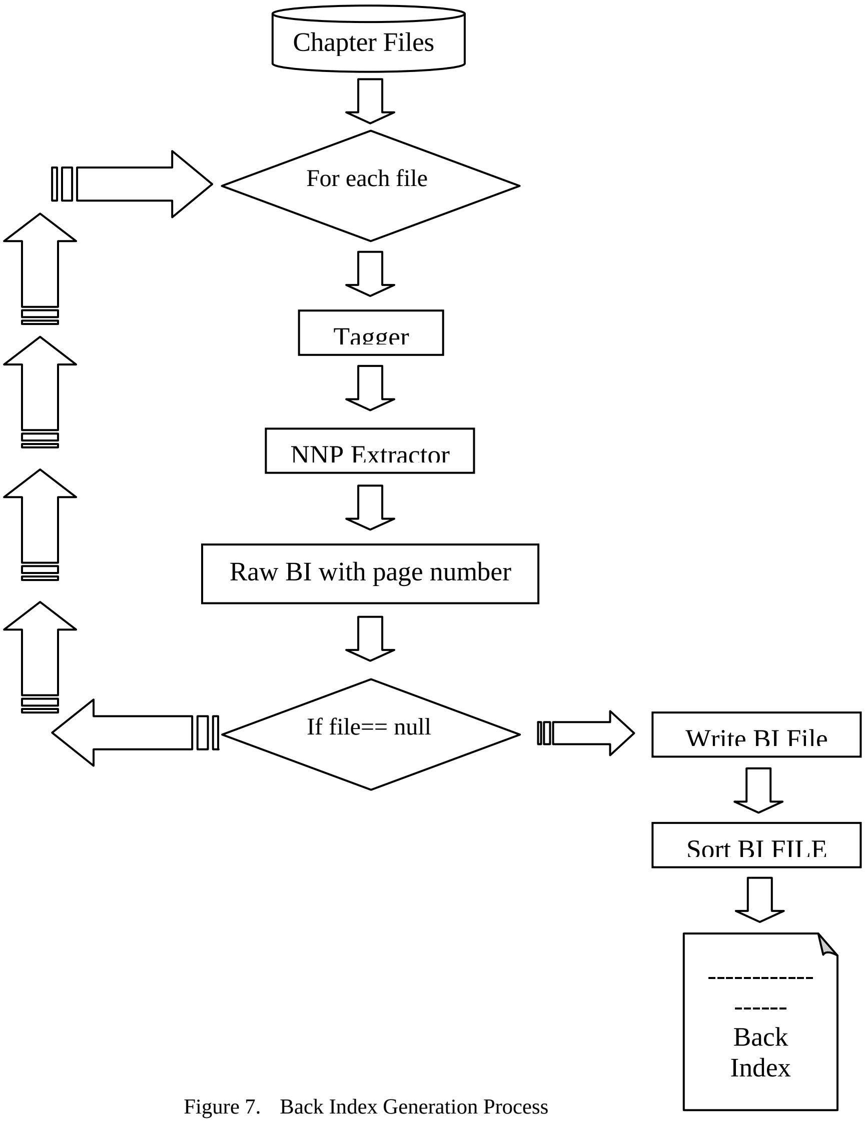
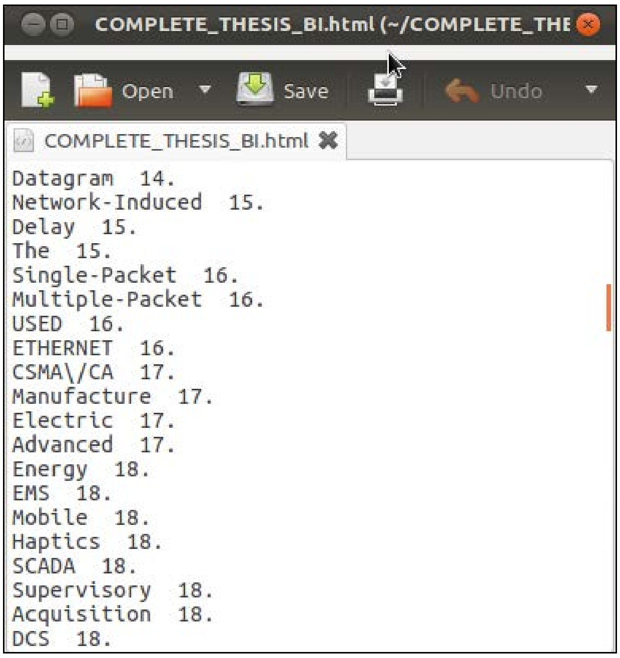


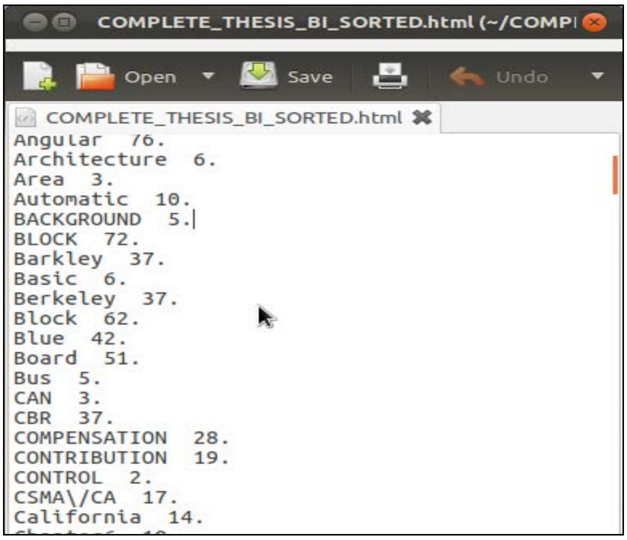
Figure 7. Back Index Generation Process

The back index generated by Meta content frame work is shown in figure 8. Back index shown in figure 8 is unsorted. In the last step a simple sorting is applied to sort the indexes. The sorted back index generated by Meta content framework is shown in figure 9.



```
Datagram  14.
Network-Induced  15.
Delay  15.
The  15.
Single-Packet  16.
Multiple-Packet  16.
USED  16.
ETHERNET  16.
CSMA\/CA  17.
Manufacture  17.
Electric  17.
Advanced  17.
Energy  18.
EMS  18.
Mobile  18.
Haptics  18.
SCADA  18.
Supervisory  18.
Acquisition  18.
DCS  18.
```

Figure 8. Un Sorted Back Index generated by Meta Content Framework


COMPLETE_THESIS_BI_SORTED.html (~/COMPI
Open
Save
Undo
COMPLETE_THESIS_BI_SORTED.html

```
Angular  76.
Architecture  6.
Area  3.
Automatic  10.
BACKGROUND  5.
BLOCK  72.
Barkley  37.
Basic  6.
Berkeley  37.
Block  62.
Blue  42.
Board  51.
Bus  5.
CAN  3.
CBR  37.
COMPENSATION  28.
CONTRIBUTION  19.
CONTROL  2.
CSMA\/CA  17.
California  14.
```

Figure 9. Sorted Back Index generated by Meta Content Framework

VIII. Results and Discussion

Several series of experiments were run to process e-books and it is observed that the time required for processing an input file depends upon the number of pages present in the book. The experimental setup was implemented upon text corpora from varied course ware (e-books in the present setup) and also with the different page counts and content sizes.

All the observations clearly indicate that the predominantly prevailing factor that contributes to excessive execution time is the varying document page size.

IX. CONCLUSIONS AND FURTHER SCOPE OF WORK

Various results of the experiments show that if the input file is in correct standard and the input file is not the OCR based PDF then the framework works well. The back index generated by the current strategy is simple and flat. Various efforts are in progress to generate the hierarchical back index.

Other aspects of back indexes are multiple occurrences handling. A single noun chunk can occur in multiple pages, the current technique arranges all the occurrences repeatedly but it should be arranged in such a manner so that single noun chunk and their page numbers, separated by commas, can be seen in a single line. The future work will focus on the generation of hierarchical back indexes with multiple occurrences handling.

X. ACKNOWLEDGEMENT

The author sincerely thanks Prof. Arpana Rawal, Prof. Ani Thomas for their timely, invaluable and indispensable guidance and consequently encouragement shown towards the work group.

REFERENCES

[1] Adobe Systems Incorporated, http://www.adobe.com/pdf/

[2] iText ® - Free / Open Source PDF Library for Java and C# , http://www.itextpdf.com/

[3] Stanford Log-linear Part-Of-Speech Tagger, http://nlp.stanford.edu/software/tagger.shtml

[4] Oracle Corporation, http://www.java.com/en/K. Elissa, “Title of paper if known,” unpublished.

[5] Sarang Pitale and Tripti sharma, “Information Extraction tools for portable document format”, International journal of computer technology and applications,Vol 2 (6), 2047-2051

[6] César García-Osorio,Carlos Gómez-Palacios,Nicolás García-Pedrajas, “A Tool for Teaching LL and LR Parsing Algorithms”, Proceedings of the 13th annual conference on Innovation and technology in computer science education, ACM New York, NY, USA ©2008, pp-317-317 M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] JL Klavans et.al. , “Computational linguistics for metadata building (CLiMB): using text mining for the automatic identification, categorization, and disambiguation of subject terms for image metadata”, Multimedia Tools and Applications, Volume 42 Issue 1, March 2009, Kluwer Academic Publishers Hingham, MA, USA.

AUTHORS PROFILE

Mrs. Tripti Sharma is Assistant Professor in Department of Computer Science & Engineering at Chhatrapati Shivaji Institute of Technology, Durg (C.G.) , India. She pursued Bachelor of Engineering from University Institute of Technology, Bhopal and Masters Degree from Rungta College of Engineering & Technology, Bhilai. She is heading the affairs of Department of Computer Science, Chhatrapati Shivaji Institute of Technology, Durg (C.G.) since 2010.Her research mainly focuses on image processing. She is a life time member of Indian Society of Technical Education, India (ISTE) and Institutional Member of Computer Society of India.

Sarang Pitale is Assistant Professor in the Department of Information Technology in Bhilai Institute of Technology, Durg (C.G.), India. He pursued his Bachelor of Engineering from Pt. Ravi Shankar Shukla University Raipur, Chhattisgarh. Currently, pursuing Masters of technology, from Chhattisgarh Swami Vivekanand Technical Unversity, Bhilai. His primary research interests focus on Data Mining techniques. He is a member of International Association of Computer Science and Information Technology, International Association of Engineers and Association for Computing Machinery (ACM).