Intelligent Parsing of Scanned Volumes for Web Based Archives

Xiaonan Lu, James Z. Wang, C. Lee Giles
The Pennsylvania State University
State College, PA 16802

Abstract

The proliferation of digital libraries and the large amount of existing documents raise important issues in efficient handling of documents. Printed texts in documents need to be converted into digital format and semantic information need to be parsed and managed for effective retrieval. In this work, we attempt to solve the problems faced by current web based archives, where large scale repositories of electronic resources have been built from scanned volumes. Specifically, we focus on the scientific domain and target scanned volumes of scientific publications. Our goal is to automate the semantic processing of scanned volumes, an important and challenging step towards efficient retrieval of content within scanned volumes. We tackle the problem by designing a machine learning-based method to extract multi-level metadata about content of scanned volumes. We combine image and text information within scanned volumes for intelligent parsing. We developed a system and test it with real world data from the Internet Archive, and the experimental evaluation has demonstrated good results.

1. Introduction

With the proliferation of digital libraries, there arise needs for efficient processing of existing printed documents. Large amount of existing documents need to be transformed into electronic formats, and content of documents need to be analyzed and managed intelligently so that digital libraries can provide users effective access to information within documents. The efficient processing of printed documents is very important for preserving and making use of historical documents.

Advancements in scanning and automatic character recognition techniques have facilitated efficient the transform from printed documents to electronic documents. Under this automated process, volumes of printed documents can be scanned and OCRed (Optical Character Recognition) by machines. This also enable digital libraries to build large repositories of electronic resources from printed documents. This automated process has become the standard for digitizing printed documents, and it also set the stage for processing semantics of scanned documents.

Although digital libraries host large repositories of scanned documents, it is difficult to provide users efficient search for content contained within scanned documents because semantics information about content may not be available. For instance, the Internet Archive [2] scan and host large amount of historical scientific publications in an attempt to provide researchers permanent access to historical collections. Volumes of scientific publications, bound in the form of book, are scanned and OCRed by machines. Every volume of printed publications is transformed into one electronic document. As a result, it is difficult for the digital libraries to provide article-level search, since there is no semantic information about content within a scanned volume such as issues, articles within each issue, etc. A user who wants to find an article may need to look through a whole volume or several volumes of publications. To achieve efficient article-level based search, automated techniques need to be designed for parsing scanned volumes intelligently.

In this work, we attempt to solve problems faced by digital libraries where scanned volumes are hosted. The goal is to parse scanned volumes intelligently and extract semantic information about content of scanned volumes. We focus on the scientific domain and target scanned volumes of scientific publications. Specifically, we aim to automate the process of identifying issues within a scanned volume, locating articles, and recording the multi-level information using metadata. These automatically generated metadata will serve as the basis for efficient and effective content-based access to scanned volumes.

Automatic metadata extraction from scanned volumes is a unique problem identified by two characteristics: there are possibly multiple issues and many articles contained within one scanned volume; texts in documents are generated from automatic scanning and character recognition. These key features make the problem different from previous related work on automatic metadata extraction from research papers [4] or title extraction from general documents [5]. In

the scenario of automatic metadata extraction from research papers, every document corresponds to one research paper. There is no need to identify the existence and location of a research paper, which is a critical step in processing scanned volumes. In the scenario of title extraction from general documents, format information (font, alignment, etc.) are proved to be effective for labeling titles. While in scanned volumes, there is no direct format information associated with texts which are generated by OCR process. Due to these differences, techniques proposed in previous works can not be applied to the problem of metadata extraction for scanned volumes. To the best of our knowledge, there is no existing work on automatic extraction of metadata from scanned scientific volumes.

It is challenging to design techniques for extracting metadata from scanned volumes automatically. There are many categories of information within a volume: cover page, table of content, notes, article, etc. Articles contained within the same volume may have different style. And, there is no embedded format information (font size, boldface, etc.) associated with text in scanned volumes. Furthermore, due to diversity across domains and along the time line, various structures are used in different fields. Finally, OCR errors in scanned volumes present another challenge. All of these make automatic article metadata extraction a difficult problem.

We combine both image and text information in parsing scanned volumes. We choose to work on the DjVu XML [1] file of scanned volumes, which is a special type of XML file for scanned documents. In the DjVu XML format, text and bounding box coordinates for every recognized word is stored. The bounding box coordinates indicate the position and size of a word within the page. By combing text and bounding box coordinates of words within scanned volume, a set of features are calculated for every line of text for the purpose of recognizing lines containing metadata elements. The line features are designed to approximate format information of texts.

The remainder of this paper is organized as follows. In section 2, we list prior work in closely related areas. In section 3, we give an overview of the whole system for processing printed scientific volumes. In section 4, we present our method for automatic metadata extraction from scanned volumes. In section 5, we describe the experimental setup and the results. Finally, we conclude our work and suggest future research directions in section 6.

2. Related Prior Work

Metadata information is the key material for the semantic web. Much effort has been put into effective generation, management, and use of metadata information, with the purpose of achieving semantic-enabled services on web.

2.1. Automatic Metadata Extraction

Automatic metadata extraction has become an important topic for digital libraries, and different techniques have been proposed to tackle specific problems in various domains. In scientific digital libraries, rule-based [3] and machine-learning based [4] methods have been proposed to extract article metadata elements for research papers. A system [12] has been developed at U.S. National Library of Medicine (NLM) to automatically generate descriptive metadata that includes title, author, affiliation, and abstract from scanned medical journals. In the educational digital library domain, methods have been proposed to extract Dublin Core and Gateway for Education (GEM) metadata from educational materials. There is also a classification-based method[5] designed for extracting titles from general documents using format features. Scanned documents raise additional issues related to previous work in document image analysis. In the case a document is represented as an image, document logical structure [13] can be derived. In another case where part of a document is represented as an image (figure), the type and other information can be obtained from content of the figure [11].

2.2. Service Oriented Structure for Digital Libraries

Metadata information generated by digital libraries not only facilitate the services provided by digital libraries but also enable efficient integration of digital libraries into the semantic web. A service oriented structure for digital libraries [14] has been proposed to integrate complex information retrieval systems into the semantic web. The integration is enabled by providing Application Programming Interface(API) for services provided by digital library systems.

Our work aims to extract metadata for a special type of documents: scanned scientific volumes. Objectives of our work include identifying logical units (volumes, issues, research papers, etc.) contained within bound volumes of scientific publications and extract metadata set for every logical unit. The extracted metadata information will enable retrieval of content within scanned volumes.

Scanned volumes of scientific documents contain much more complicated information than individual documents studied in previous works. Instead of containing a single research paper, a report, or a presentation, a scanned volume normally has a hierarchy of information including issues, articles, etc. The requirement of generating metadata sets in multiple levels makes our work distinct from related previous work.

3. Overview

The whole system for converting printed scientific volumes to searchable electronic documents is shown in Figure 1.

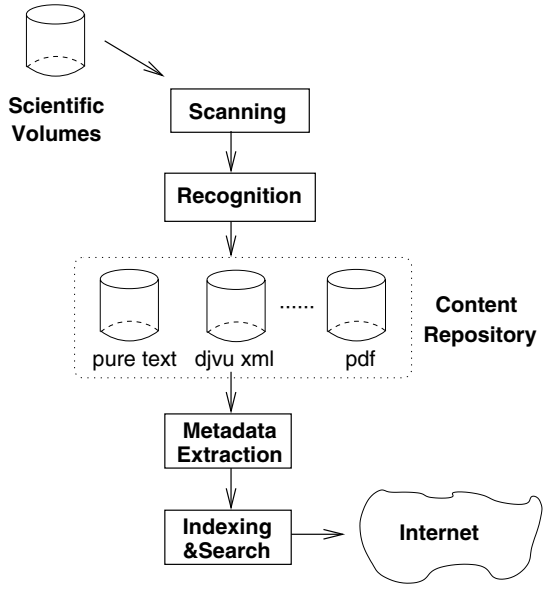

Figure 1. System architecture.

At the beginning of the system, volumes of publications are scanned and stored as images automatically. The scanned images are preserved in many different image file formats, including JPG and DjVu [1].

The recognition module reads in scanned image files and generates text information using OCR techniques. The text generated by of automatic recognition are represented in many different file formats and stored in the repository. As shown in Figure 1, txt, pdf, and DjVu XML files are generated and included in the content repository. The three types of files all contain text information and may also contain other information. For instance, the DjVu XML file records text as well as the position of words within each page; the pdf file records both text and background information for every page; and the txt file contains only text information.

Based on our goal of intelligent parsing of scanned volumes, we choose to work on DjVu XML files for metadata extraction. Even though DjVu XML consumes more storage space than pure text files, the DjVu XML file contains structured information about pages, paragraphs, lines, and words as well as the bounding box coordinate of individual words. The structured information is convenient for intelligent parsing, and most important, the bounding box coordinates of words provide position and size information about words. Combining text, position, and size information, it is possible to design line features that approximate format of text lines. The line features are designed to reflect format changes among text lines into noticeable changes in line feature values. Under this way, we use image and text information in DjVu file format to derive format features about text contained within scanned volume. Since DjVu XML format provides clues on approximating format information about text in scanned volumes, we are able to obtain both linguistic and format features from DjVu XML input files.

The metadata extraction module consists of two major components: feature extraction and metadata labeling. The feature extraction component uses text and image information to calculate line features for text lines contained within scanned volume. The metadata labeling component applies both rule-based and machine-learning based method to extract metadata about issues and articles contained within scanned volumes. Specifically, we use rule-based pattern match to detect cover pages of issues, and use machine-learning based approach to label lines containing article metadata elements. The extracted metadata includes information describing the volume and issues numbers, the article titles, article authors, the page of articles, etc. All these metadata information decomposes the content of a scanned volume into many searchable logical units.

After metadata describing scanned volumes and articles within a volume have been extracted, content of scanned volumes can be efficiently managed. Besides, multiple sources of information (document images, content repository, and article metadata) are linked together through extracted metadata. Finally, web based archives will be able to deploy search engines to provide article level retrieval to end users.

4. Metadata Extraction Method

4.1. Outline

The goal of the metadata extraction is to parse scanned volumes of scientific publications and generate descriptive metadata. The generated metadata consists of two level information: volume level and article level. The volume level information include the volume numbers and issue numbers contained in each scanned book, the total number of pages, and the start page of each issue, etc. The article level information identify all articles contained within a scanned book and extract important metadata elements including title, author, start page, etc. The extracted volume level and article level metadata reveal the content and organization of every scanned book, which serve as the basis for efficient article retrieval.

The metadata extraction method consists of two major components. One component is parsing and feature extraction, the other is metadata labeling. Figure 2 shows the

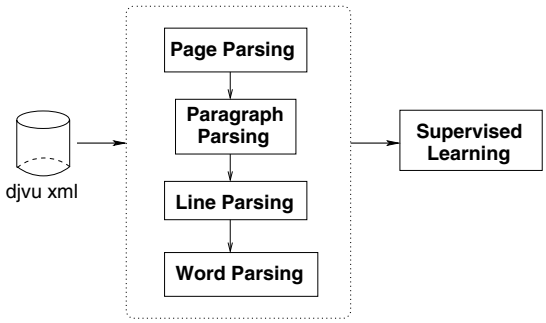

Figure 2. Metadata extraction process.

flow of functions in metadata extraction. The parsing and feature extraction part reads in DjVu XML file, parse the document and substructures inside it: page, paragraph, line, word. Based on text and image information, features are calculated for every line of text. Rule-based approach is used to detect cover pages and recognize volume and issue numbers. Supervised learning based approach is applied to identify lines containing metadata elements.

4.2. Parsing and feature extraction

The parsing and feature extraction module takes DjVu XML file as input and parse the objects contained within XML file. There are four major types of objects in every DjVu XML file: page, paragraph, line, and word. During the parsing, features about page, paragraph, and line are recorded.

```
<LINE>
<WORD coords="718,2602,1056,2546">Volume</WORD>
<WORD coords="1113,2612,1171,2553">I,</WORD>
<WORD coords="1209,2606,1349,2554">No.</WORD>
<WORD coords="1388,2608,1447,2558">1.</WORD>
</LINE>
```

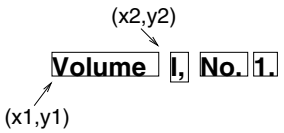

Figure 3. A sample DjVu XML line.

In DjVu XML file, there is bounding box coordinates associated with every word. For example, Figure 3 shows a sample line object. In the line object, there are four word objects. In every word object, there are text information and four coordinates $\{x_1, y_1, x_2, y_2\}$ specifying the bounding box of the word. From the bounding box information, size and position about the word can be obtained.

We choose text line as the unit for feature extraction. Compared with other objects (page, paragraph, word), line is the appropriate unit for metadata labeling. Because article title and article author usually occupy one or a few consecutive lines. And words within within the same line normally have the same format. Thus, a set of features are designed to approximate format features of text line, and feature values are calculated based on bounding box and text information. Detailed description about text line features is presented in Besides, multiple sources of information (document images, content repository, and article metadata) are linked together through extracted metadata. the following.

Font Features

Capital Mode : This feature represents the usage of capital characters in a line. The feature takes one of three possible values, $\{0, 1, 2\}$, corresponding to one of three modes: non-capital, first character capital, and full capital. The feature is designed based on our observation that it is common to use capital characters in article titles.

The capital mode of a line is dependent upon capital mode of all words within the line. For any word, its capital mode is obtained by analyzing its characters. After capital mode of all words within a line have been decided, the number of words belonging to three categories are also recorded. If we use n_0, n_1, n_2 to represent the number of non-capital, first character capital, and full capital words respectively, the capital mode of a line, represented by CM_l, is defined as the following:

$$CM_l = \begin{cases} 0 & \text{if} \quad max\{n_0, n_1, n_2\} = n_0; \\ 1 & \text{if} \quad max\{n_0, n_1, n_2\} = n_1; \\ 2 & \text{if} \quad max\{n_0, n_1, n_2\} = n_2. \end{cases}$$

Average Word Height : This feature represents the average height of words within a line. It is calculated as:

$$\frac{\sum_{i=1}^{m} \left(y_1^i - y_2^i\right)}{m}$$

where m represents the number of words in a line, y_1^i, y_2^i are bounding box coordinates of the ith word in the line. This feature is designed to approximate the height of the font used for a line.

Average Character Width : This feature presents the average width per character for a text line. It is calculated as:

$$\frac{\sum_{i=1}^{m} \left(x_2^i - x_1^i\right)}{\sum_{i=1}^{m} n_i}$$

where m represents the number of words in a line, x_1^i, x_2^i are bounding box coordinates of the ith word in the line, n_i represents number of characters in the ith word. The feature is designed to approximate the width of the font used for a line.

Normalized Average Word Height : This feature represents the relative word height of a line compared with

heights of all lines in the same page. If n represents the number of lines within a page, the normalized average word height of the ith line is calculated as:

$$\frac{\frac{\sum_{j=1}^{m_i}\left(y_1^{i,j}-y_2^{i,j}\right)}{m_i}}{\frac{\sum_{i=1}^{n}\frac{\sum_{j=1}^{m_i}\left(y_1^{i,j}-y_2^{i,j}\right)}{m_i}}{n}}$$

where m_i represents the number of words in the ith line, $y_1^{i,j}, y_2^{i,j}$ are bounding box coordinates of the jth word in the ith line. The feature is designed to compare height of a line with that of other lines in the same page.

Normalized Average Character Width This feature represents the relative character width of a line compared with character width of all lines in the same page. If n represents the number of lines within a page, the normalized average character width of the ith line is calculated as:

$$\frac{\frac{\sum_{j=1}^{m_i}\left(x_2^{i,j}-x_1^{i,j}\right)}{\sum_{j=1}^{m_i}k_{i,j}}}{\frac{\sum_{i=1}^{n}\frac{\sum_{j=1}^{m_i}\left(x_2^{i,j}-x_1^{i,j}\right)}{\sum_{j=1}^{m_i}k_{i,j}}}{n}}$$

where m_i represents the number of words in the ith line, $x_1^{i,j}, x_2^{i,j}$ are bounding box coordinates of the jth word in the ith line, $k_{i,j}$ represents the number of characters in the jth word in the ith line. The feature is designed to compare character width in a line with that in other lines within the same page.

Position Features

Alignment :This feature represents the horizontal alignment of a line within a page. It is a numerical value defined as $\frac{s_l}{s_r}$ where s_l is the distance in pixels from the left edge of a page to left side of the first character in a line, and s_r is the distance in pixels from the right of the last character in a line to the right edge of a page. The feature is designed to approximate alignment of a text line. For a text line which is central alignment in the original volume, this feature value will be close to 1.

Line Position : This feature represents the vertical position of a line within a page. It is calculated as the $\frac{y}{h_{page}}$. where y represents the vertical position of the central of a line, and h_{page} represents the height of the page in pixels. For a scanned volume which contain many articles, article titles may appear in diverse positions of a page, however, we should not expect to see article titles in header or footer part of a page.

Line Id : This feature represents the position of a line within lines in the same page. For example, the line ID of the first line is 1, the line ID for the second line is 2, etc.

Context Features

Previous Distance : This features measures the vertical distance in pixel from a text line to the previous text line in the same page.

Next Distance : This features measures the vertical distance in pixel from a text line to the next text line in the same page.

Linguistic Features

Word Count : This feature records the number of words within a line. The feature is designed based on the observation that article titles normally are not too long, and there are relatively small number of words in every title line.

4.3. Metadata labeling

A scanned volume usually corresponds to a series of issues of periodical bound in book form. In order to describe the content of a scanned volume, we define two levels of metadata: volume level and article level. Volume level metadata contain information about the whole scanned volume, and article level metadata contain information about a single article within the scanned volume.

We combine rule-based pattern matching and machine learning-based approach in the process of metadata labeling. The rule-based pattern matching method is used to label certain metadata element which has specific pattern, such as the volume number and issue number lines on cover pages of issues. The machine-learning based approach is used to detect metadata elements which may have certain types of format patterns, but it is difficult to derive the decision-rules manually.

4.4. Volume Level Metadata Labeling

The volume level metadata identify the range of issues in a scanned volume, the mapping of a page number in the original publication and the index of the page in the digitized document. In order to facilitate presentation, we use page number to refer numbers printed on the original document, and use page index to refer the index of a page in the digitized document. For instance, for a PDF file containing a scanned volume, if the PDF file has 485 pages, the range of page index is $[1, 485]$. Normally, the maximum value of page index is bigger than the maximum value of the page number for a scanned volume. Because there are empty pages, cover pages, and other pages contained within a scanned volume which do not have page numbers in the original document.

The identification of the range of issues is determined by the detection of cover page of issues. Rule-based method is used to detect special patterns in cover pages, for instance, the volume and no. information. Furthermore, duplicate cover pages for the same issue are detected, and the cover

page which is closest to the content is chosen as the start point of an issue.

In order to tolerate OCR errors in cover page detection, we use Levenshtein Distance[10] metric to measure the possible match of a string corrupted by OCR errors and a target string. For instance, for a word "volume", which is critical for cover page detection, there are some variations caused by OCR errors, such as "voluime", "volm", etc. By setting the threshold Levenshtein Distance value, we can configure the system to tolerate a specific number of character errors within one word. Based on our manual checkup, we found that there a high probability of OCR errors in cover pages, partly due to the reason that special fonts are used in cover pages and relatively more noise appear on cover pages.

Page number is recognized by analyzing text in certain areas of every page and context analysis. Since it is almost always true that page number is printed at the top or bottom area of a page, the area for searching page number is restricted. On the other hand, OCR process brings a lot of challenges. For instance, "11" on a page may be recognized as two separate words "1", and "1". To tolerate this type of problem, we combine two neighboring numbers if the spatial distance between them is below certain threshold value. As another example, number "6" may be recognized as character "b", which will cause the failure of page number detection. In order to tolerate failure in page number detection due to various reasons, we use context analysis method. Specifically, for a page without valid recognized page number, the algorithm will try to fill in the page number by checking the page number in previous and following pages.

4.5. Article Level Metadata Labeling

The goal of article level metadata extraction is to identify and describe articles contained within a scanned volume, since article search has become an essential tool in scientific digital libraries. In order to facilitate browsing and searching articles contained within scanned volumes, we aim to extract important metadata elements for every article. The current metadata element name, type, and descriptions are listed in the following Table 1.

In Table 1, several metadata items are designed to link information from different resources. For instance, the start page number, start page index, and start page image file are used to link one page in the original publication to the corresponding page in the digitized document, and the page image in the database of scanned page images. The link between different resources is designed to enable potential new features of article search.

We use a machine learning based approach to detect title and author information. Specifically, we use line as the unit and attempt to label the lines containing titles. Features of every line are extracted, as described in the previous subsection. Thus, every line in the scanned volume corresponds to one vector of feature values. For each genre of publication, we manually label a small set of lines and train the model. After that, the trained model can be used to label lines from scanned volumes of the same genre. This supervised-learning based approach is used to achieve flexibility for different academia fields, publishing authorities and diverse formatting styles.

Table 1. Article level metadata elements.

Name	Type	Description
Title	String	article title
Author	String	article author
Volume	Numerical	volume number
Issue	Numerical	issue number
Start Page	Numerical	start page number
End Page	Numerical	end page number
Page Index	Numerical	start page index
Page Image	String	start page image file

After title lines are detected, other article level metadata can be obtained based on extracted volume level metadata and title lines. For instance, since we have extracted the page (page index) of an article title, other elements such as volume number, issue number, page number, etc. can be obtained by combining extracted volume level metadata.

5 Experimental Results

5.1. Experimental System

We have developed an experimental system to parse scanned volumes and generate metadata. For the DjVu XML document parsing and line feature generation, a program is developed using Java and XML DOM parsing API. The major modules of the program are shown in Figure 2, which contains page parsing, paragraph parsing, line parsing, and word parsing. The program is designed to take DjVu XML file of a scanned volume and generate metadata in formatted text file.

In the machine-learning based title labeling, the input is a collection of vectors, where every vector corresponds to one line of text. We employ the SVM model and use SVM light [8] binary classification software to train and extract title lines. Extracted title information are combined with volume level metadata to generate the whole list of metadata elements for articles contained in the scanned volume.

5.2. Experimental Setup

We have conducted experiments on extracting metadata from scanned volumes using the proposed method. We

use precision and recall to measure the performance of automatic identification of articles contained within scanned volumes. We have collected our test dataset from the Internet Archive, an Internet library which provides access to historical collections. Tests have been conducted on selected scanned volumes of proceedings contributed by Smithsonian Institute [6].

For a test scanned volume, we manually check every page of the volume and record the information about issues and articles contained within the scanned volume. Issue numbers, cover pages, and article metadata information are recorded. These manually generated metadata information serve as the ground truth for our experiments. For a test scanned volume, we also use the DjVu XML file as input and run our experimental system to obtain automatically generated metadata information. Finally, we compare automatically generated metadata with ground truth metadata to calculate the precision and recall of article identification.

We selected two volumes of scanned books, which have about one thousand pages, and manually labeled them for our experimental test. The information about the test data is shown in the following Table 2.

Table 2. Test dataset.

Property	Value
Genre	ENTO
Volume number	{II, III}
# of issues	9
# of scanned pages	893
Time	19th century
DjVu XML file size / Volume	$7 \sim 8$M
Scanned PDF file size / Volume	$45 \sim 55$M

In the above Table 2, the "ENTO" refers to "the Proceedings of the Entomological Society of Washington". Each scanned volume in the dataset contain many articles (presented articles) with various length.

5.3. Experiments

The parsing and feature extraction module takes DjVu XML file as input, parse objects within the file, and generate feature vectors for text lines. Every vector corresponds to one line of text contained within the scanned volume. The elements in feature vector represents values of line features, which have been described in the previous section. To facilitate the presentation, the list of line features is summarized in the following Table 3.

Class labels of text lines in a scanned volume are generated from manual metadata record for the volume. Currently, we use separate indicator for title begin and title end respectively. The title begin label represents whether a line is the beginning of an article title; and the title end label represents whether a line is the end of an article title. Combining the sequence of line feature vectors generated for the scanned volume and the corresponding sequence of class labels, we obtain the experimental data for supervised-learning based title extraction.

Table 3. Line features.

Feature ID	Name
1	Capital Mode
2	Average Word Height
3	Average character Width
4	Normalized average word height
5	Normalized average character width
6	Alignment
7	Line position
8	Lind ID
9	Previous distance
10	Next distance
11	Word Count

There is an imbalance issue [7] associated with our experimental data due to the nature of our problem. For text lines contained within a scanned volume, there is only a very small ratio of title lines. For instance, in Proceedings of the Entomological Society of Washington Vol II., there are 484 pages and 3200 text lines in total. On the other hand, there are 86 articles contained in the volume which means 86 titles. Thus, the number of positive instances (title begin, title end) is much smaller than the number of negative instances, and the ratio of positive instances in the whole set is only $0.1\% \sim 0.2\%$. From previous research and our preliminary experiments, the imbalanced training set causes low performance in statistical classification process.

We adopt the idea of shrinking the majority instances [9] in order to make the dataset balanced. In our dataset, we randomly select negative instances to make the number of negative instances close to the number of positive instances. Then, a balanced data set is obtained. We use six-fold cross-validation to conduct the train and test process and collect the performance results.

5.4. Performance Results

There are volume level and article level metadata generated by our experimental system. We obtain the performance of our system on automatic metadata generation by comparing automatically generated output with the ground-truth metadata generated manually.

The performance of using rule-based string pattern match and applying Levenshtein Distance to tolerate OCR error works well on cover page detection. In our test volumes, issue cover pages and duplicate cover pages are detected successfully.

The performance of article metadata extraction is measured by precision and recall, which measure the ratio of correctly extracted metadata vs. total extracted metadata, and the ratio of correctly extracted metadata vs. total metadata respectively. Even though we tackle the labeling problem by using statistical classification method, we believe precision and recall are more precise measures than classification error rate for our problem, since we care more about the correct identification of articles and article metadata extraction. In the automatically generated article metadata output, if a set of metadata correctly identify an article, it is a success case. In our current work, the end page of an extracted article is the same page as or the previous page to the start page of the next article, depending on the location of the title lines of the next article. By manually checking the automatically generated article metadata set with the ground truth article metadata set, performance of our system on the test dataset is summarized in the following Table 4.

Table 4. Precision and Recall.

# of articles	146
# of extracted articles	141
# of correctly extracted articles	138
Precision	98%
Recall	94%

The results in Table 4 shows the effectiveness of the supervised learning based approach using the designed set of line features. It reveals that the text line features calculated based on position, size, and text information are able to approximate the format of text lines and detect format changes among text lines. And also, article title lines within scanned volumes of the same genre share certain format styles.

6 Conclusion and Future Work

We have presented our system for intelligent parsing of scanned volumes for web based archives. The whole system of handling bound scientific volumes includes scanning, automatic recognition, metadata extraction, and search. We propose a method for parsing scanned volumes, extracting text line features, and labeling volume level and article level metadata. A supervised learning-based approach is proposed for extracting metadata for articles contained within scanned volume. Both image and text information in scanned volumes is utilized for intelligent parsing. The empirical results showed that good precision and recall can be achieved on real-world use.

In the future, we plan to work on the following directions. The intelligent parsing and metadata extraction program will be integrated with other components (scanning, recognition, and search) in the document processing system. And We will apply our system to large scale repositories of scanned volumes.

This work was supported in part by the US National Science Foundation under grants 0535656, 0347148, 0454052, and 0202007, Microsoft Research, and the Internet Archive.

References

[1] Djvu zone. In *http://www.djvuzone.org/*.

[2] Internet archive. In *http://www.archive.org/*.

[3] G. Giuffrida, E. C. Shek, and J. Yang. Knowledge-based metadata extraction from postscript files. In *Proceedings of the Fifth ACM International Conference on Digital Libraries*, pages 77–84, 2000.

[4] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 37–48, 2003.

[5] Y. Hu, H. Li, Y. Cao, D. Meyerzon, and Q. Zheng. Automatic extraction of titles from general documents using machine learning. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 145–154, 2005.

[6] S. Institute. Smithsonian institute. In *http://www.si.edu/*.

[7] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. In *Intelligent Data Analysis*, pages 429 – 449, 2002.

[8] T. Joachims. Making large-scale svm learning practical. In *Advances in Kernel Methods - Support Vector Learning*, 1999.

[9] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets:one-sided selection. In *International Conference on Machine Learning*, pages 179 – 186, 1997.

[10] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, pages 707 – 710, 1966.

[11] X. Lu, J. Z. Wang, P. Mitra, and C. L. Giles. Deriving knowledge from figures for digital libraries. In *Proceedings of the International World Wide Web Conference*, pages 1229–1230, 2007.

[12] S. Mao, J. W. Kim, and G. R. Thoma. A dynamic feature generation system for automated metadata extraction in preservation of digital materials. In *Proceedings of the First International Workshop on Document Image Analysis for Libraries*, page 225, Washington, DC, USA, 2004. IEEE Computer Society.

[13] D. Niyogi and S. N. Srihari. Knowledge-based derivation of document logical structure. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 472–475, 1995.

[14] Y. Petinot, C. Giles, V. Bhatnagar, P. Teregowda, H. Han, and I. Councill. A service-oriented architecture for digital libraries. In *International Conference on Service Oriented Computing*, pages 263 – 268, 2004.