A Dynamic Feature Generation System for Automated Metadata Extraction in Preservation of Digital Materials

Song Mao, Jong Woo Kim and George R. Thoma
National Library of Medicine
Bethesda, MD 20894

Abstract

Obsolescence in storage media and the hardware and software for access and use can render old electronic files inaccessible and unusable. Therefore, the long-term preservation of digital materials has become an active area of research. At the U.S. National Library of Medicine (NLM), we are investigating the preservation of scanned and online medical journal articles, though other data types (e.g., video sequences) are also of interest. Metadata of different types have been proposed to save the information needed to preserve digital materials. Given the ever-increasing volume of medical journals and high labor cost of manual data entry, automated metadata extraction is crucial. A system has been developed at NLM to automatically generate descriptive metadata that includes title, author, affiliation, and abstract from scanned medical journals. A module called ZoneMatch is used to generate geometric and contextual features from a set of issues of each journal. A rule-based labeling module (called ZoneCzar) then uses these features to perform labeling independent of journal layout styles. However, if there are significant style variations among the issues of a same journal, the features generated from one set of journal issues may not be very useful to label a different set. In this paper, we describe a dynamic feature updating system in which the features used for labeling a current journal issue are generated from previous issues with similar layout style. This new system can adapt to possible style variations among different issues of the same journal. Experimental results presented show that the new system delivers improved labeling performance accuracy.

1. Introduction

Long-term digital preservation is a challenging problem due to several reasons including: 1) technical obsolescence of storage media and the hardware and software needed for accessing and interpreting the digital materials, and 2) the ever-increasing volume of the "endangered" digital materials and high labor cost of manual data entry. Competition in the computer industry has given rise to fast-paced releasing of new electronic file formats and their supporting hardware and software, and upgrading of existing ones. A company may choose not to support earlier versions of its electronic file formats or hardware on economic grounds. For example, Kodak recently announced the discontinuation of their Carousel slide projectors. The Library of Congress has large volumes of electronic audio files saved on old cylinders and old plastic disks that are no longer accessible. Current research in digital preservation has been focused on two major strategies: emulation and migration. In the emulation strategy, the original hardware and software environment is emulated in software so that the original digital materials are still accessible in original form with contemporary hardware and software. Lorie [1] described a Universal Virtual Computer that can emulate current hardware and software environment on future machines. However, the strategy is still in its infancy due to its complexity and very limited quantitative results have been reported. The second strategy, migration, appears to be more practical in preserving large-scale digital archives. In this strategy, digital materials are converted periodically from one format to another, from one hardware/software configuration to another, and from one generation of computer technology to another. In research conducted at NLM, the migration strategy has been considered for preservation of digital materials [2]. Metadata plays a key role in digital preservation since structural, descriptive, and administrative metadata possess the information required to migrate original digital material as well as to access it in the future [3].

While some metadata is directly encoded in the original digital material and may be automatically extracted, descriptive metadata is usually not explicitly available and is keyed in by human operators in many situations. Given the ever-increasing volume of digital materials and high labor cost, automated metadata extraction is essential. At the Communications Engineering Branch of the Lister Hill National Center for Biomedical Communications, an R&D division of NLM, a system called Medical Article Records System (MARS) [6] has been developed to automatically extract descriptive metadata such as the article title, author names, affiliation, and the abstract from scanned and online medical journal articles. These fields are then populated to MEDLINE®, the NLM's premier

bibliographic database of citations to the medical journal literature. Currently, a labeling module [4] (called ZoneCzar) is used to assign logical labels to the zones of contiguous text in scanned pages. The labeling algorithm is based on rules that depend on the layout style of the journals. However, layout style can vary greatly not only among different journal types but also within the same journal (e.g., journals with a long history tend to update their layout styles to more modern ones at periodic intervals). Style change can include changes in font size, font attributes, size, and location of individual zones. An example of style change within a journal can be found in the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (PAMI). The layout style of the regular article title page of this journal was changed in 1996. Figure 1 shows the old and current layout styles of the regular article title page in PAMI. Another example of style change can be found in the medical journal *Cerebrovascular Disease* whose style was changed in 1992 and again in 1999. The labeling rules learned from journal pages with one layout style cannot be used to reliably label the journal pages with a very different layout style and cannot generate accurate descriptive metadata. Therefore, a labeling system that is independent of layout style is necessary.

Figure 1. The old layout style (before 1996) (a) and the new layout style (1996 and later) (b) of regular papers in IEEE PAMI. Note that in the old layout style, the abstract text resides only in the left column, but in the new layout style, the abstract text runs across two columns.

In this paper, we describe a system that dynamically generates a set of geometric and contextual features to develop suitable rules. These features include font size, font attribute, and rectangular bounding box coordinates of title, author, affiliation, and abstract fields. The labeling rules are then modified to use these features to perform style-independent labeling of scanned medical journals. We then evaluate the performance of this system and compare it with the performance of our previous systems [4, 5] in which automatically generated features from new and unfamiliar layouts are not used, or features used are automatically generated from a set of pre-selected journal issues.

This paper is organized as follows. In Section 2, we present the dynamic feature updating system. In Section 3, we describe our algorithms. In Section 4, we provide an experimental design. In Section 5, experimental results are reported and discussed. Finally, a summary appears in Section 6.

2. The Dynamic Feature Updating System

The top row of Figure 2 shows a portion of the MARS system currently used to generate bibliographic data for MEDLINE. For each input page, this part of the MARS system performs the following steps:

1. The ZoneCzar1 module first generates a set of physical zones from the OCR results of the input page. It then labels the physical zones of a) the first N issues of a new journal based on rules generated from journals with regular layout styles or b) the subsequent issues of the new journal based on the journal specific features previously saved in the ZRJournalSpecificInformation table in the MARS database. The labels of interest include title, author, affiliation and abstract.
2. The Reformat module [7] rearranges the text output of ZoneCzar1 according to MEDLINE conventions.
3. Human operators use the Reconcile module to manually check the text results from the Reformat module, and correct any errors. This is the only human intervention in our system.
4. The ZoneMatch1 [5] module first matches the groundtruthed (verified) label text in the output of the Reconcile module to the zone text from the ZoneCzar module. It then generates a set of features from the matched zones of the first N issues of each new journal for title, author, affiliation, and abstract fields. Finally, it saves the generated features in the ZRJournalSpecificInformation table in the MARS table.

The dynamic feature updating system shown in the shaded area in Figure 2 is designed to generate robust features for the ZoneCzar1 module so that it can perform the labeling function independent of the layout style of the input journal. For each input journal issue, the ZoneMatch1 module is modified to first generate a set of feature distributions and saves them in ZMMriFeature

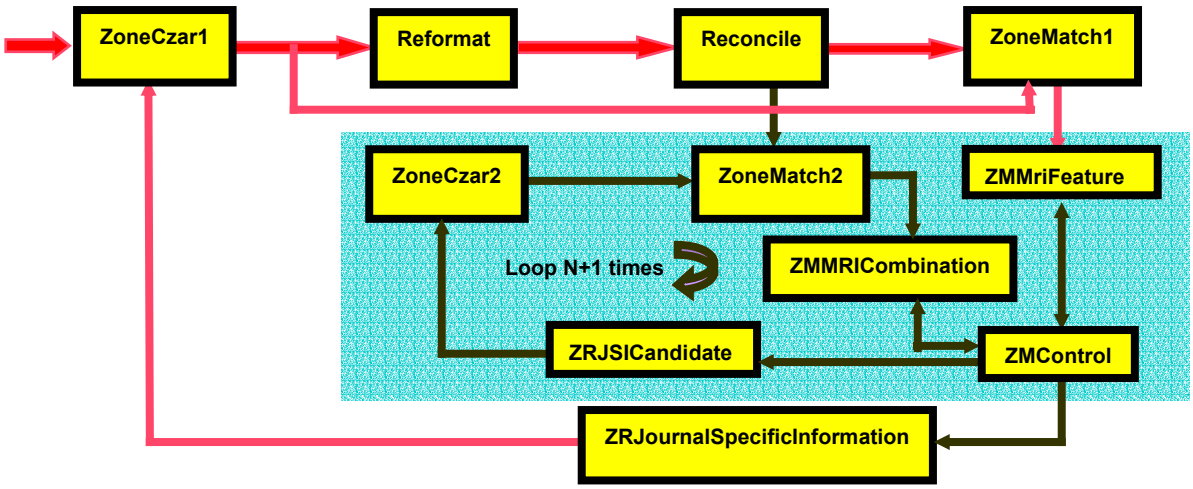


Figure 2. A dynamic feature updating system in MARS system.

table in the MARS database. The feature distributions are computed empirically from the attributes (font size, font attribute, and character bounding box coordinates) associated with the OCR output of the matched zones. Feature distribution combination means the feature distributions of a certain type (e.g., font size) generated from two or more journal issues are merged into a single distribution. Figure 3 shows font size and font attribute distributions of a journal.

The distribution of the bounding box feature is computed differently from those of font size and font attribute. For each of the title, author, affiliation, and abstract fields, we use a string-matching algorithm [5] to

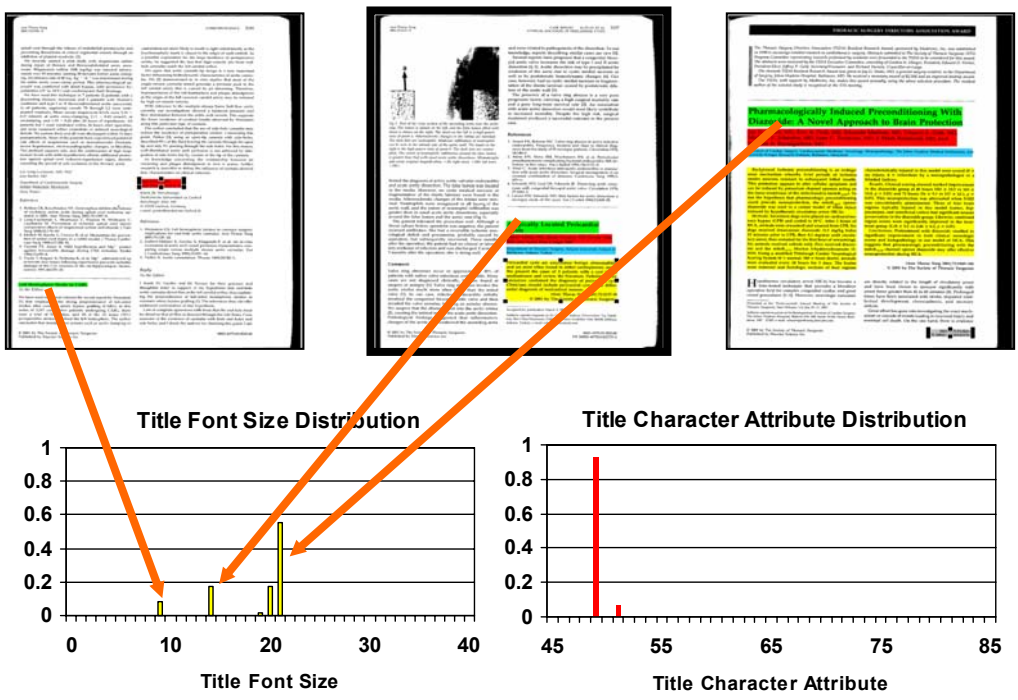


Figure 3. Font size and font attribute distributions of a journal with three layout styles. Note that font attribute is an eight bit number (0~255). Each bit represents a particular attribute such as bold face, italic, etc. The bit value of one means the font has the corresponding attribute and zero otherwise.

find bounding boxes from each article in one or more journal issues. We then group the overlapping bounding boxes into clusters. Each such cluster is considered a bounding box feature and its probability is computed as the ratio of number of original bounding boxes this cluster contains and the total number of original bounding boxes. Therefore, a generated feature set consists of three discrete feature distributions. Figure 4 illustrates how the distribution of bounding box feature is computed. We use these feature distributions in the rule-based labeling algorithm [4] to eliminate layout style dependent rules such as zone relation and location rules [5]. In the case of labeling the title zone, only zones that significantly overlap the bounding box of the title field are considered as candidates for title zones and font size and font attribute distributions of title field are used to eliminate non-title zones from the candidate zones.

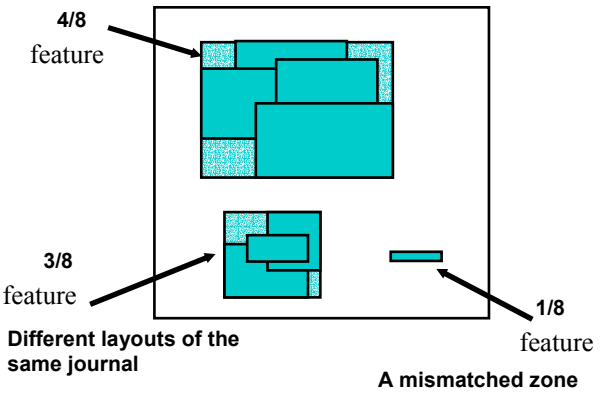


Figure 4. An example of the bounding box distribution computation.

For the currently finished journal issue, the dynamic feature updating system performs the following steps:

1. The ZMControl module first produces a combination of feature distribution sets (generated from previous issues of the same journal) in the ZMMriFeature table and then saves it in the ZRJSICandidate table.
2. The ZoneCzar2 module reads the features from the ZRJSICandidate table and generates a set of labels.
3. The ZoneMatch2 module matches the estimated labels to the corresponding verified results from Reconcile and produces a matching score.
4. Since there are many possible feature combinations, the one with the highest matching score is selected as the optimal feature set and saved in the ZRJournalSpecificInformation table. This optimal feature set will then be used by the ZoneCzar1 to label the next issue of the same journal. For example, if we need to select a combination of N feature sets from a total of $N+1$ feature sets, there are $N+1$ possible combinations of feature sets to choose from. Therefore, we need to run the labeling algorithm (ZoneCzar2) and the string matching algorithm (ZoneMatch2) $N+1$ times as shown in Figure 2.

If there is a difference in layout style among different journals or different issues of the same journal, the features generated from journals or journal issues with different layout styles will not be selected in the system. If the layout changes, the features based on the old layout, when applied to the new layout, will yield inaccurate labeling results. The Reconcile operator corrects the labels, thereby providing information for a new set of features to be generated by the matching algorithm ZoneMatch1. These will be used for subsequent issues of the journal following this new layout. Only the features generated from issues of the same journal and with a similar layout style are used to label the current journal issue. We will describe the detailed algorithm in the next section.

3. The Algorithm

We assume that the layout styles of adjacent issues (in terms of publication date) of a journal are similar except for the two adjacent issues where the layout style changes from one to another. This assumption is reasonable since even if editors of a journal decide to change its layout style drastically, the new layout style tends to stay for a long time. It is very rare that the layout style of a journal keeps changing from one issue to the next. Let $I_1, I_2, ..., I_K$ be a set of K issues (in the order of publication date) of a journal type I, and I_i is published earlier than I_{i+1}. Let F_i be a feature vector extracted from issue I_i. Let N be the number of issues to be used for extracting a combination of feature vectors. Let S_{iJ} be a score representing the labeling accuracy of the labeling module (ZoneCzar) on issue I_i using the feature vector computed as the combination of feature vectors $\{F_j, j \in J\}$. S_{iJ} is computed as

$$S_{iJ} = \frac{1}{4P_j} \sum_{m=1}^{P_j} \sum_{n=1}^{4} s(r^J_{jmn}, g_{jmn})$$

where

- r_{jmn}^{J} is the labeling result (a text string) of label type n on the title page of the article m in issue j (with P_j articles), when the feature distributions generated from the issues indexed by the elements in the vector J are used by the labeling algorithm,
- g_{jmn} is the reconciled (or groundtruthed) text string of label type n on the title page of the article m in issue j,
- $s(r_{jmn}^{J}, g_{jmn})$ is the matching score of two text strings r_{jmn}^{J} and g_{jmn}. Note that $n = 1,2,3,4$ represents the label type of title, author, affiliation, and abstract, respectively.

We use a hybrid string matching approach [5] to compute $s(r_{jmn}^{J}, g_{jmn})$. Let X and Y be two text strings with W_x and W_y words respectively and let $D(p,q;X,Y)$ denotes the minimum edit distance between X and Y up to the p^{th} and q^{th} words, respectively. $D(p,q;X,Y)$ can be computed recursively as shown in the following equation [8]:

$$\begin{aligned} D(p,q;X,Y) = \min\{ & D(p-1,q;X,Y) + C(x_p,0), \\ & D(p,q-1;X,Y) + C(0,y_q), \\ & D(p-1,q-1;X,Y) + C(x_p,y_q;X,Y)\}. \end{aligned} \quad (1)$$

where $1 \le p \le W_x, 1 \le q \le W_y$ are word indices, $C(x_p,0)$ and $C(0,y_q)$ are the preset costs for deleting word x_p and inserting word y_p, and $C(x_p,y_q;X,Y)$ is the cost for replacing word x_p by word y_p and is computed in a similar manner as in the above equation except that the basic unit is character rather than word. The optimal string matching path is obtained by backtracking in (1) starting from $D(W_x,W_y;X,Y)$. The matching score $s(X,Y)$ is then computed as the ratio of the number of the matched words in the optimal string matching path and the number of words in the reference word string, i.e.,

$$s(X,Y) = \frac{W^*}{W_y}.$$

where W^* is the number of matched words in the optimal string matching path and Y is assumed to be the reference word string. In our application, we denote the text string from Reconcile as the reference string. Note that W^* can be fractional since one word can be *partially* matched to another word if some of their constituent characters can be matched.

The steps of our algorithm for selecting features are shown as follows:

1. Initialization: compute $F_1, F_2, ..., F_N$ from issues $I_1, I_2, ..., I_N$, and let $J = \{1,2,...,N\}$. Let $i = N+1$.
2. Compute F_i from the i^{th} issue and let $J = J + \{i\}$. Create $N+1$ feature vectors $F_{e_1}^c, F_{e_2}^c, ..., F_{e_{N+1}}^c$ where $F_{e_n}^c$ is a combination of N feature vectors in $\{F_j, j \in J, j \ne e_n\}$.
3. Run the labeling algorithm on the $i+1^{th}$ issue using each of the $N+1$ feature vectors generated in Step 2. Find $e^* = \arg\max_{e \in J} S_{i(J-\{e\})}$, i.e. we find the optimal feature vector that is the combination of feature vectors computed from issues indexed by the elements in $J - \{e^*\}$. Save the optimal feature vector in the ZRJournalSpecificInformation table.
4. Let $J = J - \{e^*\}$ and $i = i+1$. Go to Step 2.

From the description of the algorithm, we see that only the feature vectors that give optimal labeling accuracy on the immediately preceding issue are used by the labeling algorithm to label the current issue. When the layout style changes among issues of a journal, the algorithm tends to keep the feature vector that is generated from previous issues with similar layout style as that of the current issue. Note that in Step 3 of the algorithm, one could use more than one issue for generating matching score and selecting optimal feature vectors.

4. Experimental Protocol

The experimental dataset includes 166 title pages from eight issues of one scanned medical journal [9] and 143 title pages from 15 issues of another scanned journal [10]. We perform our experiment once on the data from each of the two journals under the following three experimental conditions: 1) features that are

automatically generated from new and unfamiliar layouts are not used in the labeling algorithm. However, we do use some features collected from journals of several common layout styles in the labeling algorithm [4]; 2) features that are automatically generated from the first N issues are used in the labeling algorithm, and 3) features that are dynamically generated from N previous issues are used in the labeling algorithm. Average labeling accuracy over all the labeled fields of the same category (title, author, affiliation, or abstract) in the title pages of the test dataset is used as the metric for characterizing the performance of the labeling algorithm. We will compare the metric for title, author, affiliation, or abstract field and the overall metric under the above three experimental conditions. Furthermore, in the last two of the above three experimental conditions, two values of N, N=1, N=2, are used to study the labeling algorithm's sensitivity to the number of issues used for generating features.

5. Experimental Results and Discussion

Table 1 and 2 and Figure 3 show the experimental results. Examples of labeling results are given in Figure 4. Note that the experimental results are combined for the two journals.

Table 1. Labeling performance when one issue is used for generating features (N=1).

Experimental Condition	1	2	3
Title	90.45%	60.91%	94.09%
Author	88.64%	59.55%	88.18%
Affiliation	78.18%	89.09%	87.27%
Abstract	73.64%	95.91%	96.36%
Overall	82.73%	76.37%	91.48%

Table 2. Labeling performance when two issues are used for generating features (N=2).

Experimental Condition	1	2	3
Title	90.45%	77.73%	96.36%
Author	88.64%	79.09%	89.09%
Affiliation	78.18%	98.18%	92.73%
Abstract	73.64%	95.91%	98.18%
Overall	82.73%	87.73%	94.09%

We can see that the overall labeling performance under experimental condition 3 is the best one for both N=1 and N=2 even though the style variation in the journals is not very large. When N=1, the labeling performance under experimental condition 2 is the worst. This is because the number of issues used for generating features is not sufficient. Therefore, rather than helping the labeling algorithm, the inefficient features (e.g. zone bounding box) cause the labeling algorithm to miss a lot of zones. When N=2, the labeling performance under experimental condition 2 and 3 are much improved and are better than those under experimental condition 1. This is because more data (issues) are used to generate features. The labeling performance under experimental condition 1 for N=1 and N=2 do not change since features that are automatically generated are not used.

In summary, the overall and most of the categorical labeling performances of the labeling algorithm using the dynamically generated features are significantly better than those when automatically generated features are not used, or features from first N issues are used.

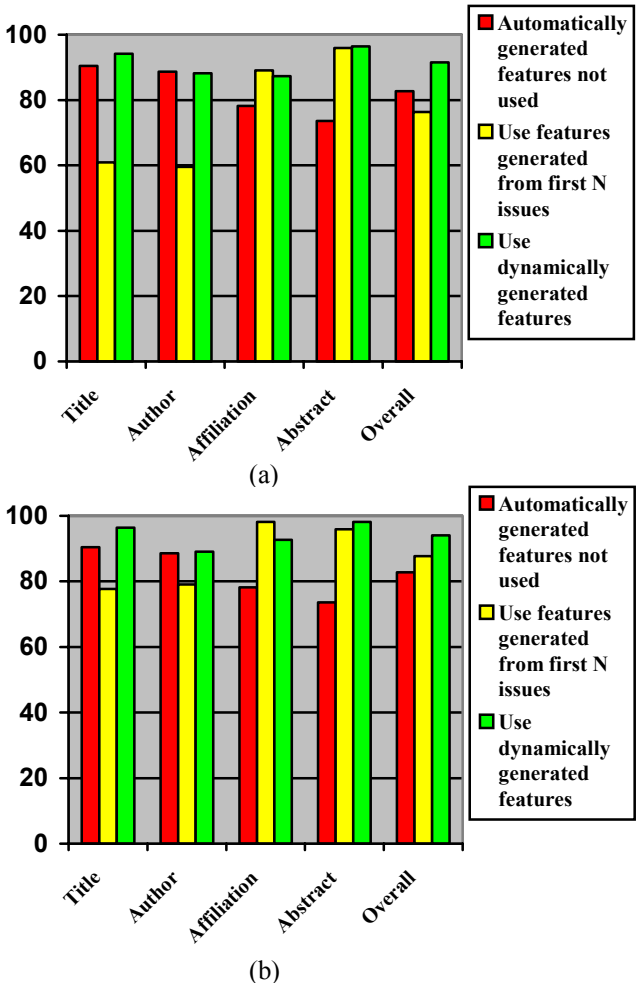
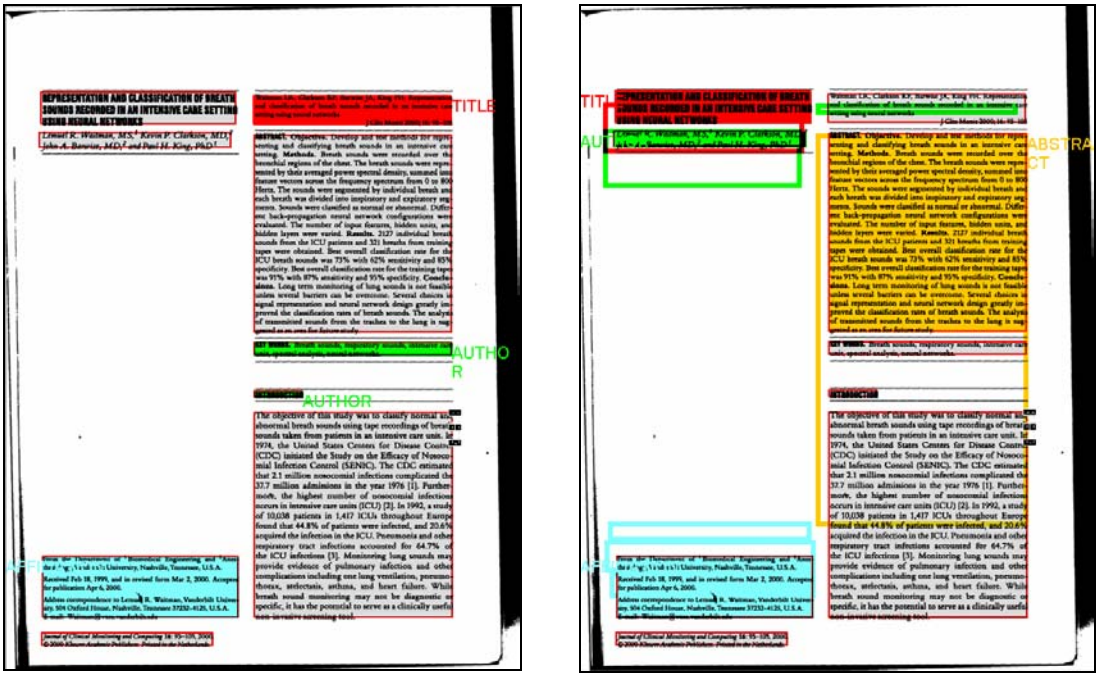


Figure 3. Labeling performance for each of the title, author, affiliation, and abstract fields and overall labeling performance for averaged over all fields when N=1 (a) and N=2 (b). Note that the values are in percentage.



(a) (b)

Figure 4. Labeling results on a title page when automatically generated features are not used in the labeling algorithm (a) and when these features are used in the labeling algorithm (b). Note that in the page shown in (a), real title and author zones are missed and other zones are mistakenly recognized as title and author zones. The real abstract zone is also missed. In (b), with the help of features (e.g. bounding boxes), the labeling algorithm successfully finds the title, author, affiliation, and abstract zones on the same page.

6. Summary

Descriptive metadata is crucial in migration-based long-term digital preservation of scanned or online journals. In this paper, we have described a dynamic feature updating subsystem to generate robust features. These features have been used in a rule-based labeling algorithm to generate accurate descriptive metadata from scanned journals independent of their layout styles. Experimental results on 309 title pages from 23 issues of two scanned medical journals have shown that the overall and most categorical labeling performances of the algorithm are best when dynamically generated features are used. The layout styles of articles in a journal can be different not only among different issues of the journal (e.g., publisher may choose to change the whole layout styles once in a while), but also among different articles in the same journal issue. For example, a journal issue can contain regular papers, short papers, correspondence, notes, etc., each of which typically has its own distinctive layout style. Currently in our system, features are generated from one or more journal issues. We plan to modify our system to generate features from one or more articles so that the system can better adapt to style changes among different articles of the same journal issue as well. We also plan to estimate the statistics of layout style changes both among different issues of the same journal type and among different articles of a same issue, and test our system on large and representative datasets. Furthermore, we will use different numbers of journal issues to generate features, and test these features in our algorithm to study their impact on labeling performance.

Reference:

[1] R.A. Lorie, Long term preservation of digital information, *Proceedings of the First ACM/IEEE Joint conference on Digital Libraries*, Roanoke, VA, June 2001, 346-352.

[2] Profiles in Science, US National Library of Medicine. http://profiles.nlm.nih.gov/.

[3] Building a National Strategy for Preservation: Issues in Digital Media Archiving, commissioned for and sponsored by the National Digital Information Infrastructure and Preservation Program, Library of Congress, April, 2002.

[4] J. Kim, D.X. Le, and G.R. Thoma. Automated labeling in document images, *SPIE Conference on Document Recognition and Retrieval VIII*, San Jose, CA, Jan. 2001, 111-122.

[5] S. Mao, J. W. Kim, and G. R. Thoma, Style-independent Document Labeling: Design and Performance Evaluation, *SPIE conference on Document Recognition and Retrieval*, San Jose, CA, Jan. 2004.To appear.

[6] G.R. Thoma, Automating the production of bibliographic records for MEDLINE, *Internal R&D report*, CEB, LHNCBC, NLM, September 2001.

[7] G. Ford, S.E. Hauser, and G.R. Thoma. Automated reformatting of OCR text from biomedical journal articles, *Proceedings of 1999 Symposium on Document Image Understanding Technology*, College Park, MD, April 1999, 321-325.

[8] R.A. Wagner and M.J. Fisher, The String-to-String Correction Problem, *Journal of ACM*, 21, 1974, 168-178.

[9] Indian Journal of Experimental Biology, 2002.

[10] Journal of Clinical Monitoring and Computing, 1999, 2000.