Style-Independent Document Labeling: Design and Performance Evaluation

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

ABSTRACT

The *Medical Article Records System* or MARS has been developed at the U.S. National Library of Medicine (NLM) for automated data entry of bibliographical information from medical journals into MEDLINE®, the premier bibliographic citation database at NLM. Currently, a rule-based algorithm (called ZoneCzar) is used for labeling important bibliographical fields (title, author, affiliation, and abstract) on medical journal article page images. While rules have been created for medical journals with regular layout types, new rules have to be manually created for any input journals with arbitrary or new layout types. Therefore, it is of interest to label any journal articles independent of their layout styles. In this paper, we first describe a system (called ZoneMatch) for automated generation of crucial geometric and non-geometric features of important bibliographical fields based on string-matching and clustering techniques. The rule-based algorithm is then modified to use these features to perform style-independent labeling. We then describe a performance evaluation method for quantitatively evaluating our algorithm and characterizing its error distributions. Experimental results show that the labeling performance of the rule-based algorithm is significantly improved when the generated features are used.

Keywords: Style independent labeling, string matching, clustering, quantitative performance evaluation, metric, geometric and non-geometric features.

1. INTRODUCTION

Bibliographic data for medical journal articles such as title, author, affiliation, and abstract is essential for both medical practitioners and researchers to retrieve relevant and reliable biomedical information. Given the ever-increasing volume of medical journals and high labor cost of manual entry, automatic data entry is necessary.

To populate the fields of MEDLINE, the flagship bibliographic citation database of the U.S. National Library of Medicine (NLM), we have developed a system, Medical Article Records System or MARS, which employs image analysis techniques to capture the article title, author names, institutional affiliations, and abstracts from the scanned pages of biomedical journals [1]. Figure 1 shows a portion of the MARS system that includes the following major components: an automatic zoning [2], and labeling [3] module (ZoneCzar), an automatic Reformat module [4], and a Reconcile module.

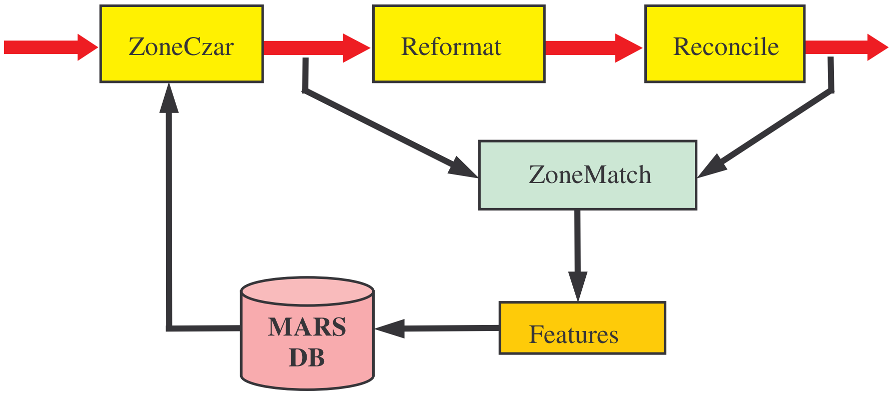

Figure 1. A portion of the MARS system used by the ZoneMatch module.

The zoning module first blocks out the contiguous text on a page image using features derived from the OCR output data. The automated labeling module then identifies the zones as the fields of interest (article title, author names, affiliations, and abstract) based on journal specific rules. Next, the Reformat module organizes the syntax of the zone contents to adhere to MEDLINE conventions (e.g., author name John A. Smith becomes Smith John A). Finally, the operators use the Reconcile module to manually correct any errors from previous automated processes. The ZoneMatch module matches the text output of the Reconcile and text of all the physical zones from the output of the ZoneCzar module, and automatically generates features from the associated OCR data of the matched physical zones for the fields of interest. The ZoneCzar module uses the generated features to replace its style-dependent rules and perform style-independent labeling. Note that the ZoneMatch module is independent of the rules used by the ZoneCzar module to assign logical labels to physical zones, it only depends on the OCR data (ASCII text, bounding box coordinates, font size, and font attribute of characters) of physical zones. Finally, a performance evaluation module is used to quantitatively characterize the labeling results of the modified ZoneCzar module against a groundtruth dataset [5].

This paper is organized as follows. In Section 2, we define the problem. In Section 3, we describe our algorithms. In Section 4, a performance evaluation method and a metric are defined. In Section 5, experimental results are reported and discussed. Finally, a summary is presented in Section 6.

2. THE PROBLEM

The automated labeling algorithm [3] depends on rules. While rules have been created for the journal titles with regular layout styles, rules have yet to be developed for titles with arbitrary layout styles (an arbitrary layout style is defined as any style that is not one of several commonly encountered regular layout styles [5]). In our current production system, new rules have to be manually created for a journal with an arbitrary layout style. Therefore, it is of interest to create a single set of rules that is independent of layout styles. In order to create such rules, we need geometric and contextual features such as bounding box coordinates, font size, and font attributes, for the important fields (title, author, affiliation, and abstract) in each journal. For example, if we know the bounding box coordinates of the title regions in the articles in a particular journal, we only need to consider the zones with the bounding box as title candidates. We can then apply a single set of rules for titles on these zones regardless of the layout style of the journal.

In the MARS system, operators use the Reconcile module to verify the text in the important fields without retaining their associated geometric and non-geometric features. On the other hand, the output of the zoning module contains both symbolic characters and their associated features such as bounding box coordinates, font sizes, and font attributes. In the next section, we will describe a module (called ZoneMatch) in which we match the output of the Reconcile module and the output of the zoning module. The features associated with the matched output of the zoning module are then saved for each of the important fields.

In order to quantitatively evaluate our algorithm, we will describe a performance evaluation methodology in which we define a computable performance metric and some error measures. The performance metric is used to represent the labeling accuracy of our algorithm and the error measures are used to study the error distributions and possible improvement of our algorithm. A quantitative metric also enables us to compare our algorithm with other approaches on common datasets.

3. THE ALGORITHMS

The core algorithms of the ZoneMatch module consist of two parts: a string-matching algorithm and a clustering algorithm for feature generation. The rule-based labeling algorithm (ZoneCzar) [3] is then modified to use the generated features to simplify its rules. The string-matching algorithm is designed to handle both merged strings as well as over-segmented strings. In the following subsections, we will describe these algorithms in detail.

3.1 The Matching Algorithm

We now describe our algorithm for matching the text outputs of the Reconcile and ZoneCzar modules. Let Σ be the character vocabulary, $C(x,0)$ denote the cost of deleting a word $x \in \Sigma^*$, $C(0,x)$ denote the cost of inserting a word $x \in \Sigma^*$, $C(x,x)$ denote the cost of exact match (usually 0) of word $x \in \Sigma^*$, and $C(x,y)$ denote the cost of substituting word $x \in \Sigma^*$ by $y \in \Sigma^*$. Let $X = (x_1, x_2, \ldots, x_M)$ and $Y = (y_1, y_2, \ldots, y_N)$ be two strings each of which consists of a set of words $x_i \in \Sigma^*, i = 1,2,\ldots,M$ and $y_j \in \Sigma^*, j = 1,2,\ldots,N$, let $D(m,n;X,Y)$ denote the minimum edit distance between the strings X

and Y up to the m^{th} and n^{th} words, respectively, and let $D(X,Y)$ denote $D(M,N;X,Y)$. If we consider a particular match between two strings as a path in the plot shown in Figure 2, each horizontal line segment represents a word insertion, each vertical line segment represents a word deletion, each diagonal line segment represents either an exact word match or a word substitution. Therefore, $D(m,n;X,Y)$ defines the optimal path and can be recursively computed as [6]:

$$\begin{aligned} D(m,n;X,Y) = \min\{&D(m-1,n;X,Y)+C(x_m,0),\\ &D(m,n-1;X,Y)+C(0,y_n),\\ &D(m-1,n-1;X,Y)+C(x_m,y_n;X,Y)\},\\ \text{where } 1 \le m \le M&, 1 \le n \le N. \end{aligned} \tag{1}$$

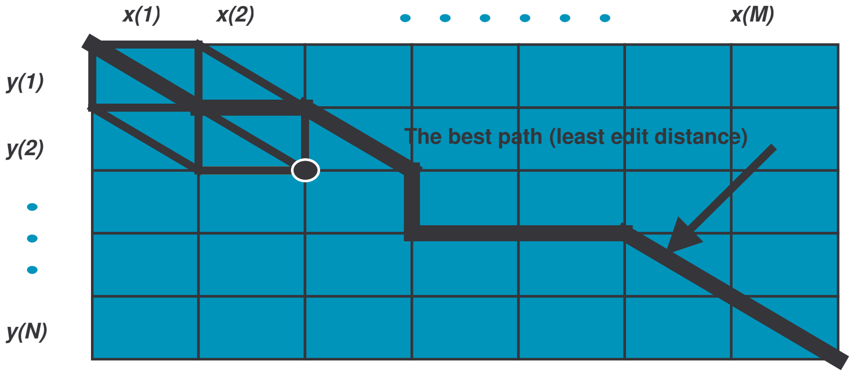

Figure 2. String-matching paths.

We compute the minimum cost of matching two words $C(x,y)$ using the same dynamic programing approach as in Equation (1). The only change in Equation (1) is that each word is replaced by a character, i.e, the problem of matching two word sequences X and Y becomes the problem of matching two character sequences x and y. Now we have a string matching algorithm that uses dynamic programming approach at both the word level and character level simultaneously. This two-level matching scheme can avoid the situation where two string of characters are optimally matched even though their underlying words do not match. For example, in a character-based string matching, a three word string *X = {abbbbbbbc, ab, bc}* is optimally mached with another two word string *Y = { ab, bc }* as follows:

```
  abbbbbbbc        ab        bc
  | |       | |
 ab        bc
```

Furthermore, it can also avoid the situation where two strings of words do not optimally match with each other even though the two strings of words only differ in a small number of characters. For example, in a word-based string matching, a two word string *X={aaaaa, bbbbb}* is not optimally matched with another two word string *Y={aaaac, bbbbe}* even though *X* and *Y* only differ in two characters, namely *c* and *e*.

We use this two-level string-matching algorithm for the title, author, and affiliation fields since they have a relatively small number of words and the characters within words tend to have OCR errors due to their font size or word length. We use the word-based string matching for the abstract fields since the abstract field has a large number of words. We will use this string-matching algorithm combined with a textline-based string matching score [8] to handle both the over-segmented zones as well as vertically merged zones (e.g. author and affiliation zones are merged together).

3.2 The Clustering Algorithm

The matched text blocks are then clustered into groups based on the minimum distance of their bounding boxes using an adjacency-list-based algorithm [7]. Only one group is selected for each field as the final match based on the following criteria:

- For title, select the matched text group with the largest font size.
- For author, select the matched text group with the largest number of characters.

- For affiliation and abstract, select the matched zone that has the closest number of words to the Reconcile text.

The distributions of three types of features (bounding box coordinates, font size, and font attribute) can be obtained from the matched data. For each journal, we compute the distributions of font size and attribute features. We also compute a probabilistic distribution of the bounding boxes of each important field in the following steps:

- Cluster the bounding boxes of each important field over all articles in the same journal that are used for feature generation.
- Count the number of bounding boxes that belong to each cluster.
- Compute relative frequency of each cluster by dividing the number of bounding boxes in the cluster by the total number of bounding boxes.
- Merge the bounding boxes in each cluster into one bounding box.

Figure 3 shows an example of computing a probabilistic distribution of bounding boxes of an important field. For example, the feature 1 bounding box contains four zones out of total eight zones; therefore its frequency is 4/8.

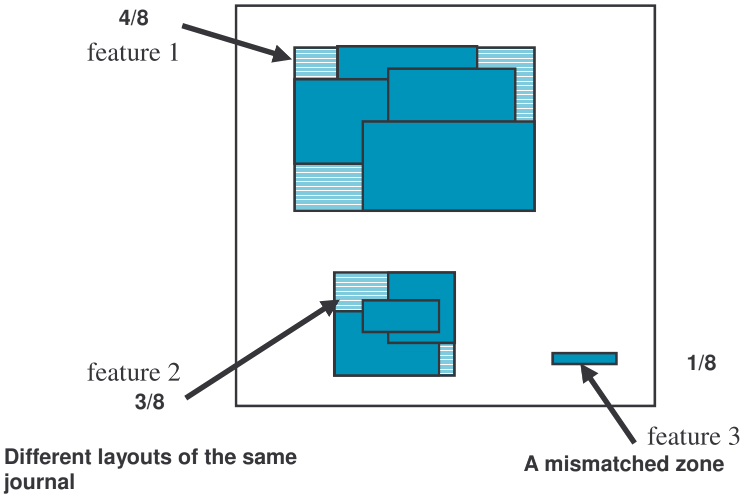

Figure 3. An example of the bounding box distribution computation.

Finally, we normalize the feature counts with respect to each page and compute the distributions of features over all pages in at most three journal issues for each journal.

3.3 Rule Simplification

The labeling (also called ZoneCzar) module consists of three types of rules: relational, zone-location, and non-geometric rules. Relational rules use geometric relationship among text zones, zone-location rules use absolute locations of zones, and non-geometric rules use text and attribute-based features. These rule types can be illustrated, as an example, in the case of labeling affiliation zones at the top of a page. Such a zone is usually located between author and abstract zones (relational rule), the zone is in the top half of the page (zone-location rule), and it contains many words suggesting affiliation, such as institution name, city, country names, etc (non-geometric rule).

For a new journal with an arbitrary layout, one has to manually create new sets of relational and zone-location rules. Since we can automatically generate features for each important field, we can use these features to eliminate relational and zone-location rules. Specifically, we can use the bounding box of each zone and consider all zones within this bounding box as labeling candidates. Only non-geometric rules are required to label these zones. The distribution of the font size feature is used to remove noisy zone candidates within the bounding box. Figure 4 shows such a rule simplification procedure.

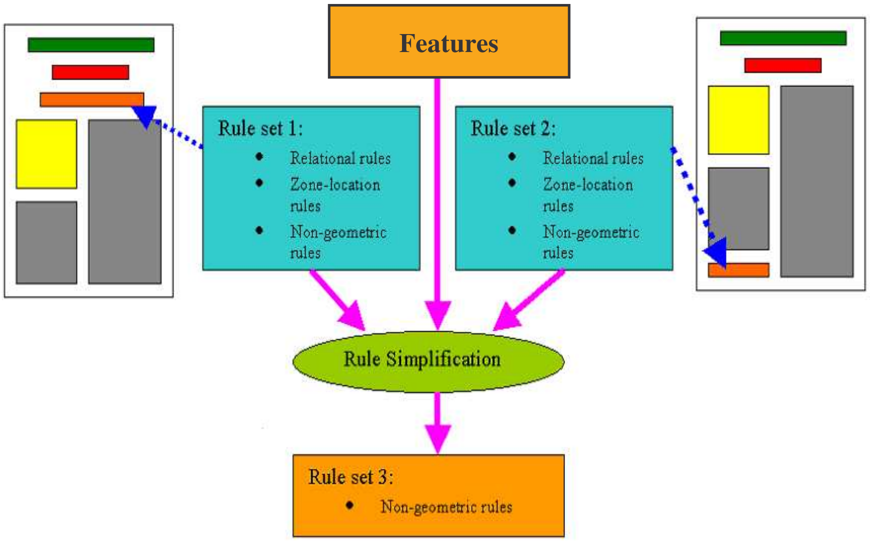

Figure 4. Rule simplification using the generated features. Note that in this figure, the rule simplification is for the affiliation field.

4. QUANTITATIVE PERFORMANCE EVALUATION METHODOLOGY

Our experimental methodology for characterizing the performance our algorithms consists of the following five steps:

- Partition the dataset into mutually exclusive training and test datasets.
- Define a performance metric.
- Compute the distributions of features on the training dataset.
- Evaluate our algorithms on the test dataset.
- Perform error analysis in different error categories

Our performance metric and error measures are based on the bounding box coordinates, label, and text content of zones. Let the page contain a set of groundtruth zones *G*; each of which has a logical label. We define four types of errors:

- False dismissals: No segmented zone significantly overlaps *G*.
- Merges: Two or more groundtruth zones significantly overlap a segmented zone.
- Cuts: *G* significantly overlaps segmented zones with different labels or overlaps with segmented zones with the same label and their complements.
- Incorrect labeling: The zone is correctly segmented (on the basis of the significant overlap), but is not labeled correctly.

Let *h* and *w* be the height and width of *G*, and let *S* be a segmented zone that significantly overlaps *G*. Significant *X* or *Y* overlap is defined in terms of a tolerance measured in percentage. We say that an *X* overlap is significant if it is at least 15% of *w*, and that a *Y* overlap is significant if it is at least 25% of *h*. We have developed a module for this evaluation purpose. Our evaluation software is based on the PSET software [9]. Our methodology is different from that presented in [10] in that a new error type, incorrect labeling, is added to reflect the logical labeling aspect of the algorithm.

5. EXPERIMENTAL RESULTS AND DISCUSSION

We first generated distributions of three types of features (bounding box, font size, and font attribute) from a training dataset of 161 page images from articles of 7 different medical journals with arbitrary layout types. We then evaluate our algorithm using the generated features on a test dataset of 49 page images from different articles of the same 7 medical journals. The training and test datasets are disjoint and taken from the groundtruth dataset described in [5]. Note that this dataset includes not only the groundtruth text and logical labels generated by human operators at the zone level using the Reconcile module but also the document images, OCR output, and operator-verified data (e.g., bounding box coordinate) at the page, zone, line, word, and character levels. We use the first three issues of each new journal to generate features. These features are then used in the labeling algorithm to label subsequent issues of the same journal. If the number of issues used for training is too small, the generated features are found to be insufficient to represent the geometric and non-geometric characteristics of the fields of interest. On the other hand, if the number of issues used for training is too great, the generated features (especially the bounding box coordinates of zones) can include many spurious or noisy features and therefore cause significant deterioration in the labeling performance. We compare the performance of our algorithm under two conditions: the generated features are, and are not, used. In Section 5.1, we report the performance evaluation results for title, author, affiliation, and abstract. In Section 5.2, we provide an error analysis of our algorithm and discuss how we can further improve its performance.

5.1 Quantitative Performance Results

Figure 5 and Table 1 show the experimental results. For each important field on each page, we first compute a labeling accuracy as the ratio of the number correctly detected and labeled zones to the number of total groundtruth zones. The correctly detected and labeled zones are defined as those zones that do not have any type of errors as defined in Section 4. We then compute an average labeling accuracy over all pages in the test dataset. Figure 6 shows experimental results on a sample page image.

Table 1. Labeling accuracy of the rule-based labeling algorithm in table representation. Note that the values in this table represent are in percentage.

Mode	Title	Author	Affiliation	Abstract
With features	97.96	95.24	57.14	92.86
Without features	95.92	41.84	36.73	48.98

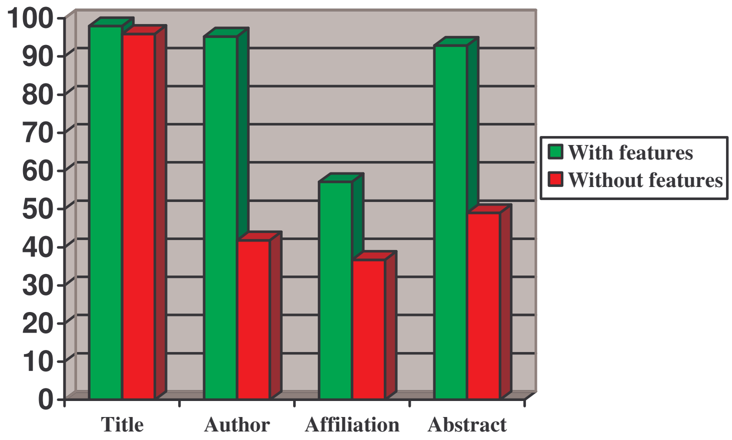

Figure 5. Labeling accuracy of the rule-based labeling algorithm under two conditions: 1) generated features are used, and 2) generated features are not used. (Note that vertical axis shows percentage.)

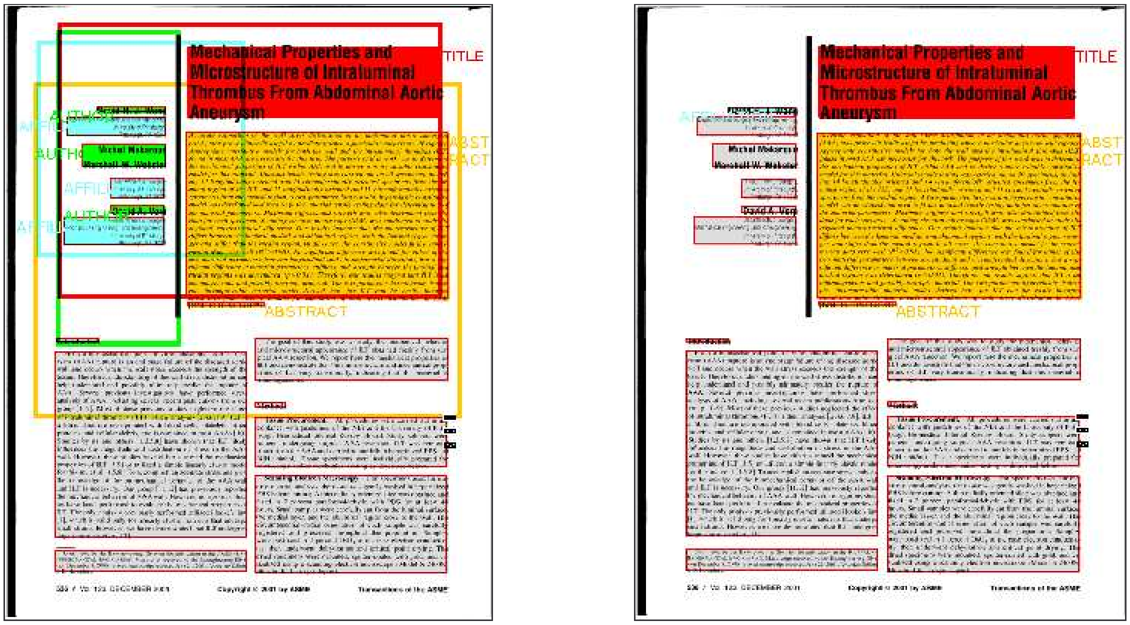
Mechanical Properties and Microstructure of Intraluminal Thrombus From Abdominal Aortic Aneurysm
TITLE
ABSTRACT
AUTHOR
AFFILIATION
Mechanical Properties and Microstructure of Intraluminal Thrombus From Abdominal Aortic Aneurysm
TITLE
ABSTRACT

Figure 6. Labeling results when the generated features are used (a) and when the generated features are not used (b).

Figure 6 shows the labeling results of the rule-based algorithm on a page when generated features are used (a) and when the generated features are not used (b). Note that title is denoted by red zone, author by green zone, affiliation by blue zone, and abstract by orange zone. The generated bounding box features are also shown. We see that all the important fields are correctly detected when the generated features are used. However, when the generated features are not used, only title and abstract fields are correctly detected, one author field is incorrectly detected as affiliation field and all other author and affiliation fields are missed.

We see that the use of generated features in the rule-based algorithm significantly improves the labeling accuracy of author, affiliation, and abstract fields. Since all title fields have the largest font size on a page, it is relatively easy for the rule-based algorithm to detect and label them even without the generated features.

5.2 Error Analysis

We compute average error rates for four types of errors: false dismissal, merge, cut, and incorrect labeling. For each page, we compute a miss error rate as the ratio of the number groundtruth zones that are not detected and the total number of groundtruth zones of all fields (title, author, affiliation, and abstract). For example, if there are four groundtruth zones (one for each of the title, author, affiliation, and abstract field) and only the title zone is not detected in a page, the miss error rate of this page is equal to ¼ = 0.25. We compute other types of errors similarly. We then compute an average error rate for each of the four types of errors over all pages in the test dataset. Figure 7 and Table 2 show the average rates for the four types of errors.

Table 2. Average error rate of the four error types. The values are in percentage.

Mode	**Missed**	**Merged**	**Cut**	**Mis-labeled**
With features	13.98	0	0.41	0
Without features	33.47	0	1.77	9.08

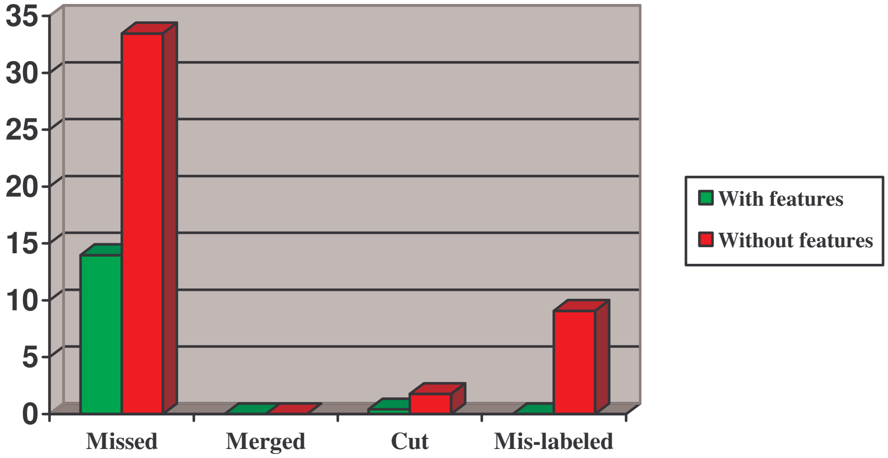

Figure 7. Average error rate of the four error types. The vertical axis is in percentage.

We see that the average error rates for false dismissal, cut, and incorrect labeling are significantly smaller when the rule-based algorithm uses the generated features. When these features are not used, the average false-dismissal and incorrectly labeled error rates are quite high. This is because the algorithm does not have information about where the important fields are located and hence uses wrong location-based rules to label them.

When the generated features are used, the average miss error rate is also high. This is because the pages in the training dataset are different from those in the test dataset and hence the bounding boxes of important fields estimated from the training dataset may not completely cover the zones in the test dataset. The labeling performance of affiliation field is significantly worse than that of other fields. This is because only the first-author affiliation is taken as the groundtruth affiliation. However, the rule-based algorithm usually detects more than just the first-author affiliation. Also, author texts in some pages are included in the affiliation text and as a result part of the affiliation texts is detected as author zones.

6. SUMMARY

In this paper we have described a module called ZoneMatch to automatically generate useful geometric and non-geometric features such as bounding box coordinates, font sizes, and font attributes. The core algorithms of this module consist of a two-level string-matching algorithm and a clustering algorithm. A rule-based algorithm was then modified to use the generated features to perform style-independent labeling. Quantitative performance evaluations of our algorithm on a groundtruth dataset have shown that the use of the generated features significantly improves the labeling accuracy of the rule-based algorithm. Our future plans are to use better features and more robust feature clustering methods to further improve the labeling accuracy of the affiliation field. We will also test our algorithm on a much larger dataset.

REFERENCES

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

2. S.E. Hauser, D.X. Le, and G.R. Thoma. Automated zone correction in bitmapped document images, *SPIE Conference on Document Recognition and Retrieval VII*, San Jose, CA, January 2000, 248-258.

3. 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.

4. 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.

5. G Ford and G.R. Thoma. Ground truth data for document image analysis, *Proc. 2003 Symposium on Document Image Understanding Technology,* College Park, MD, April 2003, 199-205.

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

7. R. Sedgewick, *Algorithms in C*, (Reading, MA: Addison-Wesley publishing company, 1990).

8. S. Mao, J. Kim, D.X. Le, and G.R. Thoma, Generating Robust Features for Style-independent Labeling of Bibliographic Fields in Medical Journal Articles, *Proc. of 7th World Multiconference on Systems, Cybernetics, and Informatics*, Orlando, FL, July 2003, To appear.

9. S. Mao and T. Kanungo, Software architecture of PSET: A page segmentation evaluation toolkit, *International Journal on Document Analysis and Recognition, 4(3),* 2002, 205-217.

10. S. Mao and T. Kanungo, Empirical performance evaluation methodology and its application to page segmentation algorithms*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(3), 2001, 242-256.