Generating Robust Features for Style-independent Labeling of Bibliographic Fields in Medical Journal Articles

Song Mao, Jongwoo Kim, Daniel X. Le, and George R. Thoma
National Library of Medicine
Bethesda, Maryland 20894, USA

ABSTRACT

Bibliographical data such as title, author, affiliation, and abstract are crucial for indexing biomedical journal articles. The *Medical Article Records System* (MARS) has been developed at the National Library of Medicine (NLM) to automate bibliographical data extraction for MEDLINE®, the NLM's premier database of citations to the biomedical literature. The automatic extraction of bibliographic data involves the process of assigning logical labels (title, author, affiliation, and abstract) to homogeneous regions or zones on page images. While an OCR- and rule-based labeling module (called ZoneCzar) in MARS can reliably label medical journals with regular layout styles, it cannot accurately label the journals with arbitrary or unusual layout styles, and new rules have to be manually created for these journals. Furthermore, the OCR zoning errors, particularly merging errors, can greatly affect the labeling accuracy of ZoneCzar. In this paper, we describe an algorithm for automatic generation of robust features that are used by the labeling algorithm to perform style-independent labeling.

Keywords: MARS, MEDLINE, style-independent labeling, rule-based algorithm, string matching.

1. INTRODUCTION

In MARS [1], an OCR- and rule-based labeling module (called ZoneCzar [2]) has been developed to automatically label title, author, affiliation, and abstract zones in the page images of medical journal articles. While the module can handle pages with regular layout styles, it cannot reliably label those with arbitrary or atypical layout styles. The regular layout styles [3] typically encountered are shown in Figure 1. An "arbitrary" layout style is defined as any style that is not one of the styles shown in Figure 1. Figure 2 shows two examples of document pages with arbitrary layout styles. New rules have to be manually created for page layouts of these types. In this paper, we describe a module (called ZoneMatch) for automated generation of robust geometric and non-geometric features such as the distributions of font size, font attribute, and bounding box from raw OCR data for each of the important fields (title, author, affiliation, and abstract) to be extracted. The ZoneCzar module then uses these features to perform style independent labeling.

Figure 3 shows a portion of the MARS system that includes a labeling module [2], a reformat module [4], and a reconcile module. The labeling module, called ZoneCzar, labels the zones created by a previous page segmentation stage. The Reformat module modifies the content of the labeled zones according to MEDLINE® conventions. Finally, text verification operators use the Reconcile module to correct any errors from previous modules, thereby generating groundtruth symbolic text for each important field.

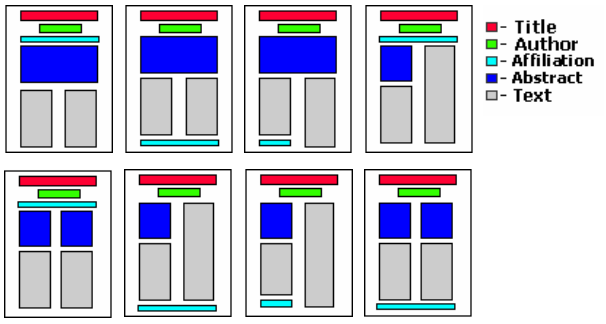

Figure 1. Regular layout styles encountered in biomedical journals.

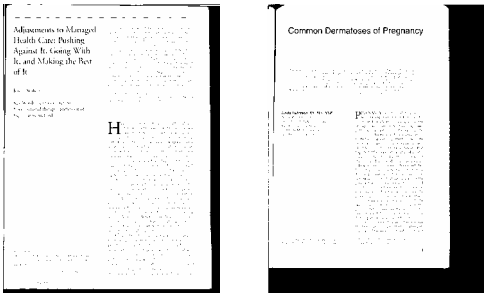

Figure 2: Document pages with arbitrary layout styles.

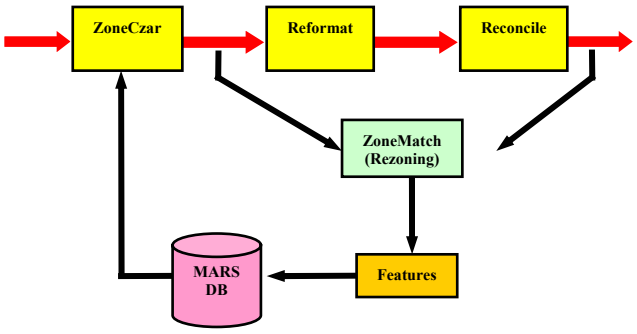

Figure 3. A portion of the MARS system used by the ZoneMatch module to generate robust features.

While the output of the Reconcile module is groundtruth text, it does not contain certain features such as font size, font attribute,

and bounding boxes since operators do not verify these features in the normal operation of the MARS system. On the other hand, the output of the ZoneCzar module contains these features in addition to the text. The ZoneMatch module matches the outputs of ZoneCzar and Reconcile to generate robust features. This module consists of two major functions: string matching and feature distribution computation. In the following sections, we will first describe these functions in detail, and then explain how the ZoneCzar module uses the generated feature distributions to perform style-independent labeling. Since the prior zoning step is not ideal, there are merged zones as well as over-segmented zones. We will introduce a string-matching algorithm that can split the vertically merged zones as well as merge the over-segmented zones.

This paper is organized as follows. In Section 2, we describe our string-matching algorithm. In Section 3, a feature generation procedure is proposed. A rule-simplification process is described in Section 4. Finally in Section 5, we report experimental results and discuss future work.

2. STRING MATCHING

In order to extract features for important fields (title, author, affiliation, and abstract) from article pages, we match the groundtruth text from the output of the Reconcile module with the raw text from the output of the ZoneCzar module. Figure 4 shows an example of the matching procedure. We base our string-matching algorithm on edit distance of strings (deletion, insertion, and substitution) and a dynamic programming approach [5].

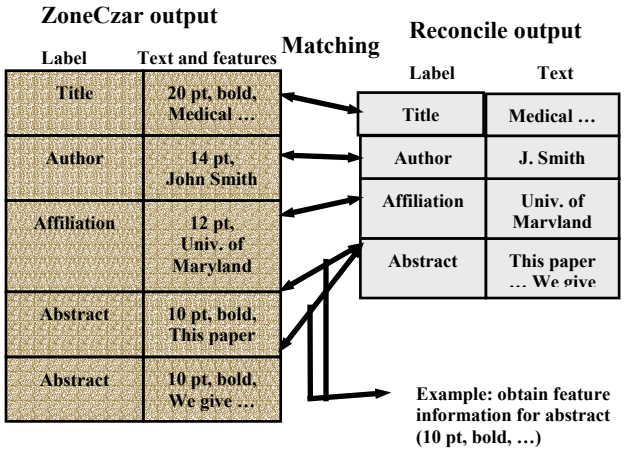

Figure 4. String-matching procedure.

Let Σ be the character vocabulary, $C(a,0)$ denote the cost of deleting a character $a \in \Sigma$, $C(0,b)$ denote the cost of inserting a character $b \in \Sigma$, $C(a,a)$ denote the cost of exact match (usually 0) of character $a \in \Sigma$, and $C(a,b)$ denote the cost of substituting character $a \in \Sigma$ by $b \in \Sigma$. Let $X = (x_1, x_2, \mathrm{K}, x_M)$ and $Y = (y_1, y_2, \mathrm{K}, y_N)$ be two strings where $x_i \in \Sigma, i = 1,2,\mathrm{K}, M$ and $y_j \in \Sigma, j = 1,2,\mathrm{K}, N$, let $D(m,n;X,Y)$ denote the minimum edit distance of the strings X and Y up to their m^{th} and n^{th} characters, respectively, and $D(X,Y)$ denote $D(M,N;X,Y)$. If we consider a particular match between two strings as a path in the plot shown in Figure 5, each horizontal line segment represents an insertion, each vertical line segment represents a deletion, each diagonal line segment represents either an exact match or a substitution. Therefore, $D(m,n;X,Y)$ defines the optimal path and can be recursively computed as

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

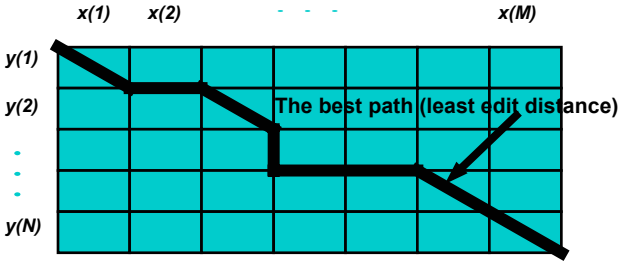

Figure 5. String-matching paths.

We call this algorithm a character-based string-matching algorithm. This algorithm becomes a word-based algorithm if the unit of input strings X and Y in (1) is word instead of character. Furthermore, this algorithm becomes a hybrid algorithm if we use the character-based algorithm to compute minimum cost of word substitution in a word-based algorithm. We use the hybrid string-matching algorithm for title, author, and affiliation fields and the word-based string-matching algorithm for abstract field. This is because title, author, and affiliation fields usually have a relatively small number of words and the characters in those words tend to have errors due to incorrect OCR conversion, and abstract field usually has a large number of words.

Due to non-ideal OCR zoning, zones encompassing different fields may be merged together, and/or the zone for a single field may be over-segmented into multiple sub-zones. From our observations, the majority of merging errors are in the vertical direction. This is especially the case for author and affiliation fields as shown in Figure 6a. An example of split errors is shown in Figure 6b. Merged and over-segmented zones are the major sources of errors in feature generation for important fields. We now formally describe a new string-matching algorithm that can handle both of these zoning errors.

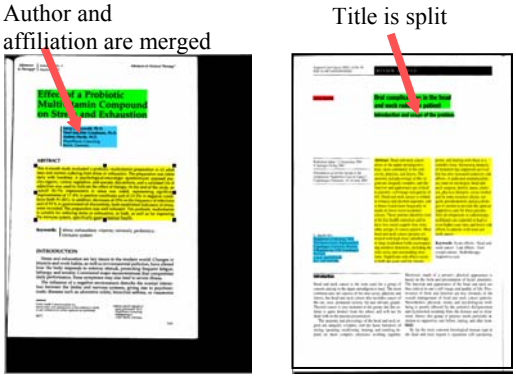

Figure 6. Zoning errors: merge (a) and split (b).

2.1 Split Vertically Merged Zones

We first describe the part of the string-matching algorithm that can split vertically merged zones. Let string T be the groundtruth text string (from Reconcile module) of an important label, let X be an input string (from ZoneCzar module) that is the content of an input zone. From the associated OCR data, we can express X in terms of a set of sub-strings each of which corresponds to a text-line, i.e., $X = (l_1, l_2, \text{K}, l_n)$. Let d_i denote the number of matched characters in d_l for the best matching path defined by $D(T, l_i)$. Let I be a subset of the index set $I_0 = \{1,2,\text{K}, n\}$. We want to maximize the following matching score within a certain threshold R with respect to I:

$$s(I) = \frac{1}{|I|}\sum_{i \in I}\frac{d_i}{|l_i|} \qquad (1)$$

We start with $s(I_0)$ where all lines of the input string X are used for its computation. We remove a pair (d_i, l_i) with the least value from Equation (1). We stop this process until the score in Equation (1) is greater or equal to R. The algorithm is described as follows:

Input: R, $I = (1,2,\text{K}, n)$ and $\{(d_i, l_i), i \in I\}$.

Output: an index set I that is a subset of $\{1,2,\text{K}, n\}$ and s.

Procedure:

1. $a = \min\left\{\frac{d_i}{|l_i|}, i = 1,2,\text{K}, |I|\right\}$,

   $i^* = \arg\min\left\{\frac{d_i}{|l_i|}, i = 1,2,\text{K}, |I|\right\}$.

2. $s = \frac{s \cdot |I| - a}{|I| - 1}$.

3. If $s < R$, $I = I - \{i^*\}$, go back to step 1, else stop.

If $s \geq R$, we split the input zones into those containing the remaining lines and those containing the removed lines. In our algorithm, we use a threshold of R = 0.83 for title, author, affiliation, and abstract fields. Figure 7 shows the rezoning result of a page with merging errors.

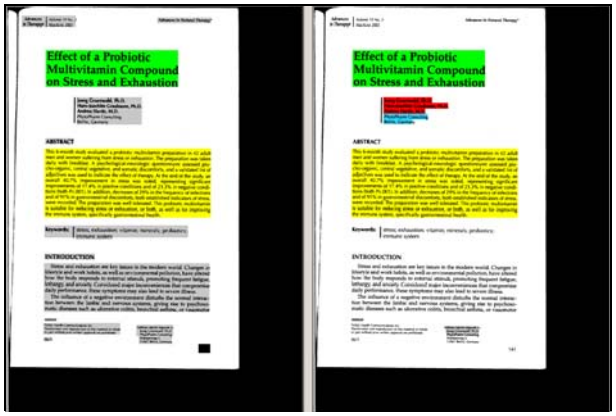

Figure 7. Rezoning of merged zones. Left page shows original OCR zoning result and right page shows the rezoning result using the above algorithm. Note that author and affiliation are correctly segmented in the rezoning result.

2.2 Merge Over-segmented Zones

We now describe the part of the string-matching algorithm that merges the split zones. Since the OCR zoning results are not ideal, the true text of an important field can be matched to more then one zone if the true zone of the field is over-segmented. We cluster the matched zones based on the minimum distance of their bounding boxes using an adjacency-list-based algorithm [6] and a distance threshold. One or more zone clusters may be generated and only one of them is selected as the final matched zone. The selection criteria are based on observed general features of the four important fields and are shown in Table 1.

Table 1. Zone selection criteria.

Label	Selection Criterion
Title	Zone cluster with the largest font size
Author	Zone cluster with the largest number of characters
Affiliation and abstract	Zone cluster with the closest number of words to the groundtruth text

Figure 8 shows an example of the clustering process, and a real page where two zones are matched to the groundtruth title text, and two zones are matched to the affiliation true text. Since the two matched title zones are close, they are merged into one zone cluster and selected as the final matched zone for the title field. On the other hand, the two matched affiliation zone candidates are far apart, and only the top one is selected as the final matched zone for the affiliation field, according to the criteria stated above.

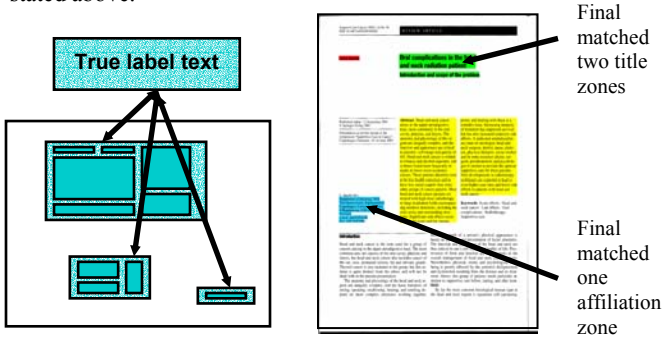

Figure 8. Clustering of split zones.

3. GENERATION OF FEATURE DISTRIBUTIONS

For the important fields (title, author, affiliation, and abstract) in each article of a journal, we compute distributions of three types of features: font size, font attribute, and bounding box. Following this computation, we normalize the distributions with respect to the total number of characters in the matched zones. We finally add these distributions over all first pages of the articles to generate final distributions for the journal. For the bounding box distribution, we merge the bounding box of the matched zone of each page over all pages in the journal into bounding box clusters based on the minimum distance of the bounding boxes. As shown in Figure 9, there are eight bounding boxes and three clusters; the probability of each cluster is computed as the fraction of the bounding boxes in the cluster to the total number of bounding boxes so that the probabilities are 3/8, 4/8, and 1/8.

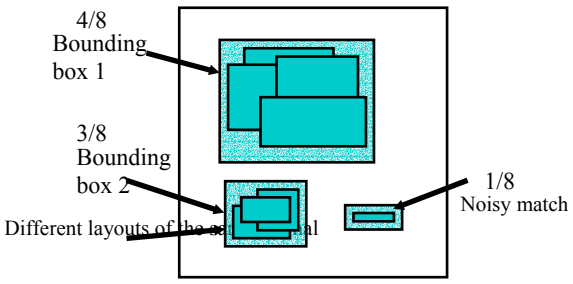

Figure 9. Bounding box clustering.

4. RULE SIMPLIFICATION USING GENERATED FEATURES

The ZoneCzar module uses the generated bounding box distributions to eliminate style dependent rules, namely, zone location and relational rules. Only non-geometric rules based on the key words and text characteristics of important fields are used to label the zones. Figure 10 shows the rule elimination process for the affiliation field. Since we generated bounding box feature for affiliation fields, we can search possible affiliation candidates within the bounding box. The zone location and relational rules about different affiliation fields in the two journals are eliminated.

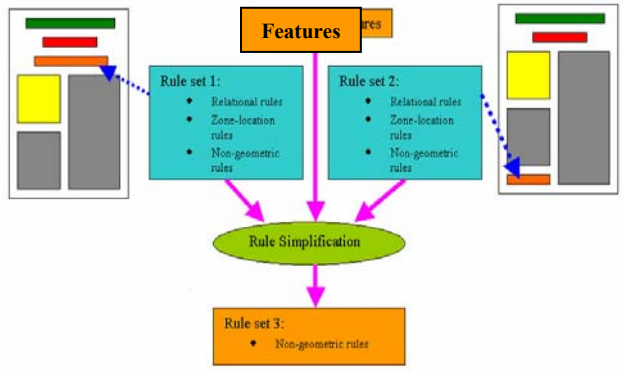

Figure 10. Rule simplification for affiliation field.

5. RESULTS

As shown in Table 2, experiments on 1,342 pages from 117 journals with arbitrary layout styles show that the labeling results of the ZoneCzar module are significantly improved when the generated features are used.

Table 2. Labeling accuracy of the ZoneCzar module.

	Title	Author	Affiliation	Abstract
With features	98.06	92.32	93.29	92.62
Without features	93.44	57.08	77.65	84.50

For each journal, we have used at the most three issues to generate the feature distributions. These feature distributions are then saved in database tables that are read by the ZoneCzar module. Figure 11 shows the labeling results of the ZoneCzar module on a page when generated features are not used (a) and when the generated features are used (b). Since the bounding boxes of important fields are generated, only the zones that significantly overlap with the bounding boxes are considered as possible labeling candidates. The distributions of font size and attribute are used to remove noisy zone candidates. We plan to use more features (such as the percentage of capital letters) and better string-matching algorithms. We will also train and test our algorithm on larger datasets with arbitrary layout styles.

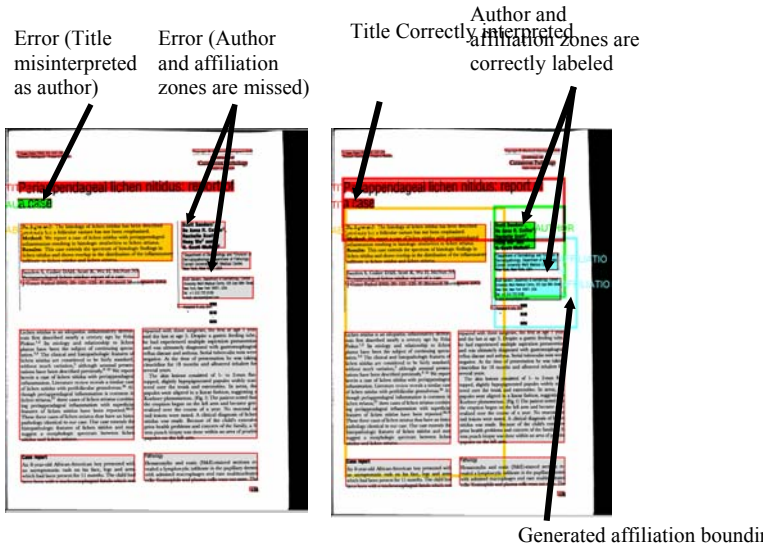

Figure 11. Labeling results when generated features are not used (a) and when generated features are used (b).

REFERENCES

1. Thoma GR. Automating the production of bibliographic records for MEDLINE. Internal R&D report, CEB, LHNCBC, NLM; September 2001; 92.

2. Kim J, Le DX, Thoma GR. Automated labeling in document images, SPIE Conference on Document Recognition and Retrieval, San Jose, CA, Jan. 2003. SPIE vol. 4307, pp. 111-22.

3. Ford G. and Thoma GR, Ground Truth Data for Document Image Analysis, SDIUT 2003, pp. 199-205.

4. Ford G, Hauser SE, Thoma GR. Automated reformatting of OCR text from biomedical journal articles. Proceedings of 1999 Symposium on Document Image Understanding Technology, April 1999, College Park, MD, pp. 321-5.

5. Wagner RA and Fisher MJ. The String-to-String Correction Problem. Journal of ACM, vol. 21, pp. 168-178, 1974.

6. Sedgewick R, Algorithms in C, Addison-Wesley publishing company, 1990.