Extracting Author Meta-Data from Web using Visual Features

Shuyi Zheng[1] Ding Zhou[1] Jia Li[2] C. Lee Giles[3]
[1]Department of Computer Science and Engineering [2]Department of Statistics
[3]College of Information Sciences and Technology
Pennsylvania State University
{shzheng, dzhou}@cse.psu.edu jiali@stat.psu.edu giles@ist.psu.edu

Abstract

Enriching digital library's author meta-data can lead to valuable services and applications. This paper addresses the problem of extracting authors' information from their homepages. This problem is actually a multiclass classification problem. A homepage can be treated as a group of information pieces which need to be classified to different fields, e.g., Name, Title, Affiliation, Email, etc. In this problem, not only each information piece can be viewed as a point in a feature space, but also certain patterns can be observed among different fields on a page. To improve the extraction accuracy, this paper argues that visual features of information pieces on a homepage should be sufficiently utilized. In addition, this paper also proposes an inter-fields probability model to capture the relation among different fields. This model can be combined with feature-space based classification. Experimental results demonstrate that utilizing visual features and applying the inter-fields probability model can significantly improve the extraction accuracy.

1. Introduction

Authors are kernel objects for any online digital library. Most available information of an author is extracted from his/her online documents, e.g., a research paper [5]. This approach has two limitations. First, only a few fields are available in an author's online document. Second, such information tends to be out of date.

In this work, we attempt to extract author meta-data from their homepages. We choose homepage domain as the extraction source it is more reliable, comprehensive and up-to-date than any other sources.

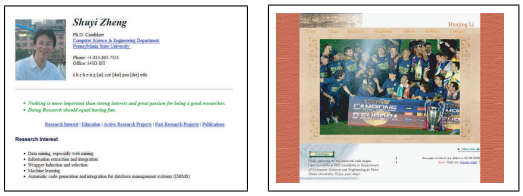
If we could extract authors' relevant information from their homepages, not only this information can serve as meta-data for a digital library, but also many valuable services and applications can be developed based on the enriched author information. Here are some typical examples. 1) Author's meta-data can be used in ranking authors and their publication documents for a digital library. 2) Enriched author meta-data would help a lot to distinguish multiple authors with same name. 3) Based on extracted author meta-data, we can even build a vertical search engine [13] to search scholars over the world.

However, extracting information from homepages is not an easy task because most homepages are manually designed and encoded. People design their homepages in all kinds of ways, which leads to the diversity of their web appearance. Traditional machine learning methods encounter difficulty when dealing with homepage domain because features found on some pages might not applicable on others.

In this paper, we proposed to take advantage of visual features to help improve the extraction accuracy. In some sense, visual features are more stable than content-based features for certain types of information which an author puts on his/her homepage. This is understandable because when people design their homepages, they usually will follow some hidden conventions of the way they put certain types of information on the page.

Due to the noisy nature of homepage data, solely depending on individual features is still hard to achieve high extraction accuracy. In this work, we propose to consider the relation of different fields on a same page. Actually, when people design their homepage, they follow some conventions not only for design some individual fields, but also on how to arrange multiple fields on a same page. This observation gives us a hint that we could learn a probability model to simulate the relation between different fields. That is what we call inter-field probability model in this paper.

However, with all above ideas taken into account, it is still a hard problem to find a uniform method to handle all kinds of different cases. In order to apply our method, we first need to limit the range where our approach is applicable. By observation, most homepages can be grouped into two types. In the first type, most (if not all) relevant information of the author has been presented on the first page



(a) Type-1 (b) Type-2

Figure 1. Two Types of Author Homepage

(e.g., index.htm); while, in the second type, author information is evenly broken into several categories. Each category has its own page. Normally, the first page of this type only conveys a little self-introduction or greetings. In terms of proportion, type-1 homepages is the majority of all homepages. And our work is focused on dealing with the type-1 homepage and leave type-2 as future work.

The rest of this paper is organized as follows: Sec. 2 reviews related work. Sec. 3 describes some fundamentals of our approach and gives a formal definition of the problem we are trying to solve here. Sec. 4 explores features of individual elements and presents an algorithm to extract author mata-data. Sec. 5 describes our inter-fields probability model. Experimental results are presented in Sec. 6. We conclude the paper in Sec. 7.

2 Related Work

We discuss the related work from two points of view.

First, our work is also a multiclass classification problem. Considering relations among different fields/classes is not totally new. Some previous work already tried to utilize such information to improve the classification accuracy. A typical example would be *Conditional Random Field*(CRF) [10]. It is a type of discriminative probabilistic model most often used for the labeling or parsing of sequential data, such as natural language text or biological sequences.

Second, Our work belongs to the area of Web Information Extraction (IE). It is receiving a lot of attentions in last years. We group these works into two categories and briefly describe each class as follows. We refer the reader to a good survey [9] and two tutorials [15][11] for more works related with IE.

Extracting structured data in template level: The most popular technique for this category is wrapper induction. Several automatic or semi-automatic wrapper learning methods have been proposed. Typical representatives are WIEN [8], SoftMeley [6], Stalker [12], RoadRunner [3], EXALG [1], TTAG [2], work in [14], ViNTs [19] and work in [21].

Vision assisted techniques: The problem of judging the importance or function of different parts in a web page attracts a lot of attentions. In proposed approaches [7, 16, 17, 18, 20], visual features, such as width, height, position, etc., are reported useful.

3 Data Representation and Problem Definition

3.1 Visual Block and Visual Tree

A Web page can be parsed into a DOM tree. Each DOM node correspondes to a *visual block*. Each block is a visible rectangular region on the Web page with fixed position and nonzero size. It is rendered from a pair of HTML tags or text between them [20]. Apparently, all visual blocks also form a tree structure.

We take page in Figure 1(a) for instance. Its DOM tree structure of the top part of the page is displayed in Figure 2 (Deep level blocks are omitted).As we can see, the whole page can be considered as the biggest block (called page level block) rendered by `<BODY>` and `</BODY>`.

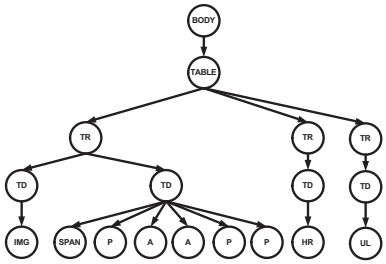


Figure 2. DOM tree for page in Figure 1(a)

Given a visual block b, we use $Width(b)$, $Height(b)$, $Left(b)$ and $Top(b)$ to indicate its size and position respectively where $(Left(b), Top(b))$ is the coordinate of left-top corner of b.

For evaluation convenience, we assign a unique identity for each block withn a Web page, call *Block ID*.

3.2 Problem Definition

The input of our problem is a HTML page p and a predefined author ontology O. p can be parsed into a visual tree consisting of a set of visual blocks, namely B. Note that each block b is a node on the visual tree. Ontology O defines a set of fields, in this paper,

$$O = \left\{ \begin{array}{c} \textit{name, title, affiliation, address} \\ \textit{picture, email, telephone, fax} \end{array} \right\}$$

Given B and O, the problem of extracting author metadata is actually a multi-classification problem. We need to design a function $Predict : B \rightarrow O$, such that for each leaf block $b \in B$, $Predict(b) = l$, where $l \in \{None\} \cup O$.

If $l = None$, it means the input block belongs to no class.

Since it is a classification problem, some state-of-the-art methods can be used to train the classifier. In our implementation, we use adaboost because it is an effective algorithm in automatic feature selection.

4 Extraction Method

In this paper, we take a tradition machine learning approach to address the problem. Our method consists of two phases: training and extraction. First, Adaboost [4] is used to train a binary classifier for each field. Then, all these binary classifiers are combined to predict the label of a given block.

4.1 Visual features

One characteristic of our method is that we take advantage of visual information for extraction.

1. **Position fetures**: *Left*, *Top*, *CenterX*, *CenterY*, *RatioOfCenterXCenterY*[1], *RelativeToPageLeft*[2], *RelativeToPageTop*, *RelativeToPageCenterX*, *RelativeToPageCenterY*, *RatioOfCenterXPageWidth*[3], *RatioOfCenterYPageHeight*, *Nested Depth*
2. **Size features**: *Width*, *Height*, *Area*, *RelativeToPageArea*, *RelativeToPageWidth*, *RelativeToPageHeight*,
3. **Shape features**: RatioOfWidthHeight
4. **Rich format features**: *FontSize*, *RelativeToPageFontSize*, *FontStyle*[4], *FontWeight*, *RelativeToPageFontWeight*, *TextAlign*, *Visibility*, *IsHyperLink*

There are three things worth noting in above feature definition. First, some features are calculated based on other basic features, e.g., *area*. Second, some features are relative features whose values are calculated based on the different between a block iteself and the root block (page). Third, some features will be converted to a numerical number before used for training, e.g., normal *FontStyle* is converted to 0 and italic *FontStyle* is converted to 1.

4.2 Content based features

Besides visual information, 70 context-based features are used to capture those subtle characteristics of different fields. Defining features based on context text is not new and has already been widely used in all kinds of content-based classification problems.

[1] *CenterX*/*CenterY*

[2] $this.Left - page.Left$

[3] $this.CenterX/page.Width$

[4] normal, italic, oblique

4.3 Algorithm of the prediction function

Here we present the algorithm used to predict a label l of a input block b.

Algorithm: Predict(b)

```
1.  begin
2.    Load 8 binary classifiers C = {c_i}, (i ∈ O)
3.    l := None
4.    maxScore := 0.0
5.    foreach c_i in C
6.      score := S(c_i, b)
7.      if score > maxScore then
8.        maxScore := score
9.        l := i
10.     endif
11.   endforeach
12.   return l
13. end
```

Figure 3. Algorithm for predicting label of a input block

Note that, in line 6, function $\mathcal{S}$ returns a float value indicating the confidence of predicting block b as label i. This value is obtained by normalizing the original prediction score of binary classifier c_i to range (-1, 1). The final label is associated with the binary classifier whose score is a maximal positive number. If all prediction scores are negtive, then the input block will be labeled as *None*.

5 Apply Inter-Fields Visual Correlation

Due to the miscellaneity of homepage domain, method proposed in previous section has several issues. 1) It is hard to distinguish fields whose feature vectors are very close. For example, address and affiliation sometimes "look" similar. 2) It tends to produce fault-positive label for blocks which are not describing the author itself, e.g., an collaborator's email address.

Another contribution of our work is that we propose to address above issues by considering the visual correlation between different fields. The idea is natural. By observation, we find that most people do not put those fields randomly on their homepage. They tend to arrange them in a way which is widely used by other professionals. For instance, a large number of people will put Name above Title and place them together beside a picture, as illstrated in Figure 4.

Based on such observation, we can define a probability model to let the inter-fields visual relation play a role in determining the label of a block. Here, the basic idea is that when a block b_1 is labeled as a certain field l_i, another

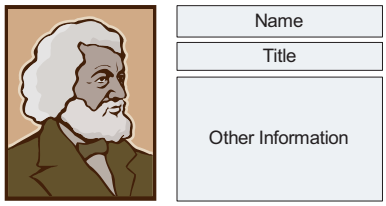


Figure 4. A typical homepage layout

block b_2 who has a certain visual relation with b_1 have a conditional probability of being another field l_j.

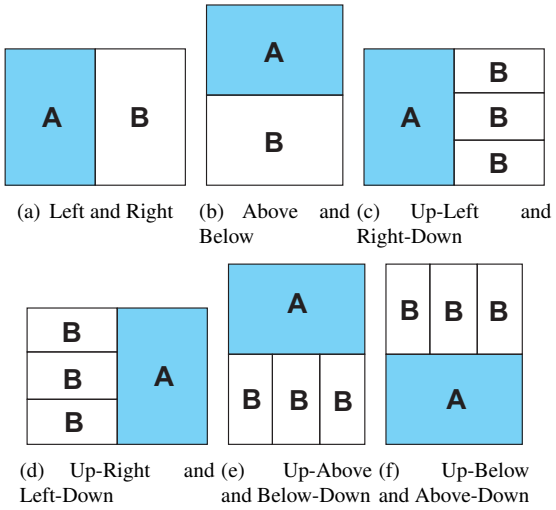
In this section, we first formally define 12 types of visual relation. Then we describe our probability model and discuss how those parameters can be learned.

5.1 Twelve Types of Visual Relations

Except a few extreme examples (less than 1%), an inner block is cut into several child blocks either horizontally or vertically. We define all visual relations based on this assumption. Those exceptional pages are excluded before feed to our extraction process. For description convenience, we use *h-block* to denote an inner block which is horizontally segmented and use *v-block* to denote an inner block which is vertically segmented.

Based on blocks' placement relation and nested depth on visual tree, twelve types of visual relations (indicated as set R) are defined. They are: *Left, Right*, *Above, Below*, *Up-Left, Right-Down*, *Up-Right, Left-Down*, *Up-Above, Below-Down*, *Up-Below, Above-Down*.

Note that we use "up" and "down" to denote the relation in terms of nested depth and use "above" and "below" to denote the placement relation at same depth.



(a) Left and Right (b) Above and Below (c) Up-Left and Right-Down

(d) Up-Right and Left-Down (e) Up-Above and Below-Down (f) Up-Below and Above-Down

Figure 5. Twelve Types of Visual Relations

Given any two blocks b_1 and b_2 on a same page, we use function $\mathcal{R}(b_1, b_2)$ to return their visual relation. It can also return *None* if no above relation exist between b_1 and b_2. All above relations are displayed in Figure 5. Each sub-figure shows $\mathcal{R}$(A, B) and $\mathcal{R}$(B, A).

Now, the question is how to implement function $\mathcal{R}$ for any two input blocks. Actually, visual relation can be easily inferred from the visual tree structure of given input blocks. For example, Figure 6 is the corresponding visual tree structure of Figure 5(c). In this figure, v-block and h-block are distinguished by a symbol inside their corresponding nodes.

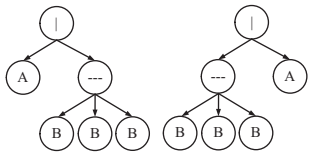


Figure 6. Visual Tree Structure of Figure 5(c) and Figure 5(d)

Given a block b and a visual relation r, from the visual tree we can also find a block set $\mathcal{N}(b, r)$ where all blocks have relation r with b.

$$\mathcal{N}(b, r) = \{b' | \mathcal{R}(b, b') = r\}$$

5.2 Inter-Fields Probability Model

Here we define the conditional probability model to express how inter-fields relation contributes in predicting the label of a given block. Suppose we are given that a block b_1 is labeled as l_i, and $\mathcal{R}(b_1, b_2) = r$, then the conditional probability of b_2 being l_j is

$$P(b_2 \in l_j | b_1 \in l_i, \mathcal{R}(b_1, b_2) = r)$$

In this model, we do not distinguish feature difference among different blocks, which means above probability is independent with b_1 and b_2 themselves. Thus we can simply use $P(r, i, j)$ to represent above probability.

Now we are in a place to define a function $\Phi(b, l)$ to calculate the confidence of labeling a block b as field l solely based on the inter-fields model where defined here.

$$\Phi(b, l_j) = \sum_{r \in R} \sum_{b' \in \mathcal{N}(b,r)} \sum_{i \in O, \mathcal{S}(b', c_i) > 0} \mathcal{S}(b', c_i) \cdot P(r, i, j)$$

Here $\mathcal{S}(b', c_i)$ serve as a weight and is directly calculated by the binary classifier c_i.

Now we can get two confidence value for any block being labeled as a certain field. One is returned by the corresponding binary classifier, the other is returned by the inter-fields model.

Natually, the best solution is to combine these two values with a smoothing factor $\alpha \in (0, 1)$. Thus, we have a final confidence functon $\mathcal{S}'$ to replace $\mathcal{S}$ (line 6, Figure 3) in function $Predict$.

$$\mathcal{S}'(b, c_i) = \alpha \cdot \mathcal{S}(b, c_i) + (1 - \alpha) \cdot \Phi(b, l_i)$$

We call the new function with $\mathcal{S}'$ as $Predict'$.

5.3 Learning Parameters

One parameter in our model is $P(r, i, j)$ for all possible combinations of relations and fields. They form a $12 \times 8 \times 8$ joint probability matrix. This matrix can be learned from manually labeled traning samples simply by the definition of conditional probability.

$$P(r, i, j) = \frac{\left\|\{(b_1, b_2)|\mathcal{R}(b_1, b_2) = r, b_1 \in l_i, b_2 \in l_j\}\right\|}{\left\|\{(b_1, b_2)|\mathcal{R}(b_1, b_2) = r, b_1 \in l_i\}\right\|}$$

$(r \in R, i, j \in O)$

In our implementation, another parameter, the smoothing factor α, is set to a fixed value 0.4 according to experience.

5.4 When No Pattern Is Found

Although our work is based on the assumption that people follow certain conventions when they design their homepages, it is still not a surprise to see weird designs since they are written manually. For an unconventional homepage, it is imaginable that inter-fileds model might not help, which means $\Phi(b, l)$ is close to zero.

6 Experimental Results

6.1 Data Collection

We collected our homepage data by meta-search. First, we preapred a list of author name obtained from online digital library *CiteSeer*. Then, we throw each name as a query to Google and get a list of urls. Using several heuristic rules, we choose one url which is most likely to be the homepage of the input query.

6.2 Evaluation Method

500 pages are randomly selected from our homepage repository. All pages are manually labeled for extraction result evaluation. Evaluation for each page is performed by comparing manually assigned labels with automatically assigned labels in our extraction process. We use precision, recall, and F1 as measures. A three-folder cross-validation is implemented for the sake of fairness.

We implemented two groups of experiments on a PC with 3 GHz Pentium 4 processor and 1015MB RAM.

Experiment I: The Power of Visual Features

The objective of the first experimental group is to show that utilizing visual features greatly improves the extraction accuracy. We implemented two sets of classifiers using different features. For the first set, all features are used to train the classifiers. For the second set, only content based features are used. Both classifier set are used to perform extraction on the same data set. In both experiments, we use algorithm $Predict$ described in section 4 without applying the inter-fields model. For the sake of convenience, we use set-1 and set-2 to denote these two sets.

Figure 7 shows the comparison results of using visual features in training and not using visual features in training. As displayed in Figure 7, although set-2 outperforms set-1 for extracting telephone and fax, for most fields, using visual features greatly improves the extraction accuracy. Especially, using visual features improve the F1-Value by 49.9% for name field and 52.2% for picture field. This can be explained as follows. For picture field, the only non-visual feature can play a role is the tag-name (`IMG`) since we do not analysis pixelss. Therefore, without considering their position and size, there is no way to tell which different `IMG` is an author's picture and which `IMG` is an unrelated. For name field, normally, there will be more than one name mention on a homepage. Solely based on context text, it is hard to distinguish those names. With the help of visual features, name of the author itself can be easily identified.

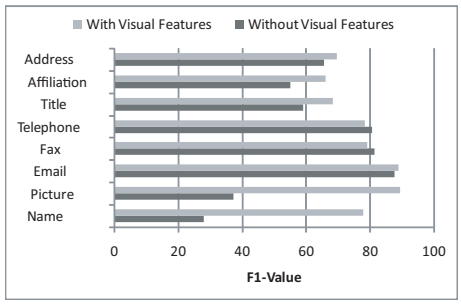


Figure 7. Result of experiment one

Experiment II: Effectiveness of Inter-Fields Model

The aim of this group of experiments is to demonstrate the effectiveness of our inter-fields model. Two extraction algorithm $Predict$ and $Predict'$ are tested on the same data set. The comparison results are shown in Figure 8. As we can see, applying inter-fields model leads to improvement for most fields. For address and affiliation, the extraction accuracy is improved by around 10% in terms of F1-Value.

The reason why inter-fields model works more effective on these two fields than other fields is that these two fields are very similar and are hard to tell from each other without considering the inter-fields relation.

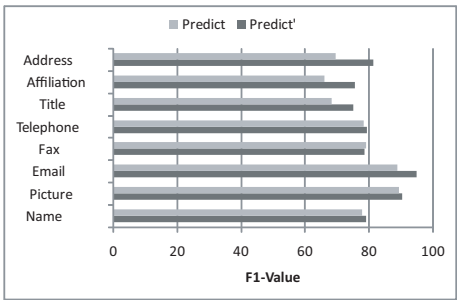


Figure 8. Result of experiment two

We also list detailed results of $Predict'$ in Table 1. The average accuracy of eight fiels in terms of F1-Value is 81.92%.

Field	Precision	Recall	F1-Value
Name	75.28	83.72	79.28
Picture	92.01	89.14	90.55
Email	94.08	95.89	94.98
Fax	82.67	74.88	78.58
Telephone	80.62	78.33	79.46
Title	74.24	76.19	75.2
Affiliation	80.46	71.52	75.73
Address	77.82	85.71	81.57

Table 1. Result of meta-data extraction

7 Conclusions

This paper addresses the problem of extracting author's meta-data from their homepages. We sufficiently utilizes visual features and applies an inter-fields probability model to capture the relation among different fields. Experimental results demonstrate that utilizing visual features and applying inter-fields probability model can significantly improve the extraction accuracy.

References

[1] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *Proceedings of SIGMOD-2003*.

[2] S.-L. Chuang and J. Y.-j. Hsu. Tree-structured template generation for web pages. In *Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence*, 2004.

[3] V. Crescenzi, G. Mecca, and P. Merialdo. Roadrunner: Towards automatic data extraction from large web sites. In *Proceedings of VLDB-2001*.

[4] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[5] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. Fox. Automatic document metadata extraction using support vector machines. In *Proceedings of JCDL-2003*.

[6] C.-N. Hsu and M.-T. Dung. Generating finite-state transducers for semi-structured data extraction from the web. *Information Systems, Special Issue on Semistructured Data*, 23(8):521–538, 1998.

[7] M. Kovacevic, M. Diligenti, M. Gori, and V. Milutinovic. Recognition of common areas in a web page using visual information: a possible application in a page classification. In *Proceedings of ICDM-2002*.

[8] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper induction for information extraction. In *Proceedings of IJCAI-1997*.

[9] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2):84–93, 2002.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-2001*.

[11] B. Liu. Web content mining (tutorial). In *Proceedings WWW-2005*.

[12] I. Muslea, S. Minton, and C. Knoblock. A hierarchical approach to wrapper induction. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, 1999.

[13] Z. Nie, J.-R. W. Wen, and W.-Y. M. Ma. Object-level vertical search. In *Proceedings of the Third Biennial Conference on Innovative Data Systems Research*, 2007.

[14] D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of WWW-2004*.

[15] S. Sarawagi. Automation in information extraction and data integration (tutorial). In *Proceedings of VLDB-2002*.

[16] R. Song, H. Liu, J.-R. Wen, and W.-Y. Ma. Learning block importance models for web pages. In *Proceedings of WWW-2004*.

[17] X. Yin and W. S. Lee. Using link analysis to improve layout on mobile devices. In *Proceedings of WWW-2004*.

[18] X. Yin and W. S. Lee. Understanding the function of web elements for mobile content delivery using random walk models. In *Special interest tracks and posters of the 14th International Conference on World Wide Web*, 2005.

[19] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu. Fully automatic wrapper generation for search engines. In *Proceedings of WWW-2005*.

[20] S. Zheng, R. Song, and J.-R. Wen. Template-independent news extractioin based on visual consistency. In *Proceedings of AAAI-2007*.

[21] S. Zheng, R. Song, D. Wu, and J.-R. Wen. Joint optimization of wrapper generation and template detection. In *Proceedings of SIGKDD-2007*.