Visual HTML Document Modeling for Information Extraction

Radek Burget

Brno University of Technology, Faculty of Information Technology,
Bozetechova 2, 612 66 Brno, Czech Republic
burgetr@fit.vutbr.cz

Abstract. Current methods of information extraction from HTML documents are mostly based on the discovery of some patterns in the HTML code that are expected to identify a particular information in the document. However, this approach has several common problems that are caused mainly by the great variability of HTML and related technologies and the fact that the HTML constructions have no direct binding to the data semantics. We propose an alternative information extraction method that is based on modeling the visual appearance of the document. Since the visual appearance of the text and the document layout have an important role in presenting the data to the reader, they have much stronger relation to the semantics of the presented data. For this reason, the proposed method has certain interesting features in comparison to the currently used methods.

1 Introduction

Current World Wide Web presents a generally accepted source of information from a broad range of areas. However, the loose form of the data presentation using the HTML language and the absence of the semantic information make any automatic processing of the contained data a non-trivial task.

Current methods of information extraction from HTML documents are mostly based on the discovery of some patterns in the HTML code that are expected to identify a particular information in the document. However, this way of mapping the semantics of the document content to the patterns of the HTML code has no support in the HTML language itself and the creators of the documents don't expect it either. The purpose of HTML and the aim of the page creators when using it is to define the *visual appearance* of the document, not the semantics of the content. Therefore, the way of the HTML usage is only limited by reaching the desired design of the page. On the other hand, there are strict limits regarding the ways of presenting the information to the reader. There exist some widely accepted typographical rules and conventions that must be respected if the document is to be readable. For example, the headings are written with larger font size than the remaining text, words in bold are more important, etc. Since the HTML documents are usually designed to be quickly understandable, these rules give the reader significant and often unique information about the structure of the document.

In this paper, we present a new approach to information extraction. The idea can be sumarized in following points:

– Creating an abstract model of the visual properties of the document that is independent on the used technology and the way of its use (particular HTML tags or CSS rules). We call this model a *visual information model*.
– Creating a model of the *logical structure* of the document based on the interpretation of certain features of the visual information model.
– Finding some patterns in the logical structure model that correspond to the information that should be extracted.

The main advantage of this approach is that the patterns are not discovered in the HTML code directly but in some abstract model. This should make the method independent on the way how the presentation has been created.

2 Related Work

This approach combines three areas that have been intensively investigated separately.

In the information extraction area, the largest part is dedicated to the automatic wrapper generation. A large group of methods is based on grammatical inference [4] and on inductive learning [2].

The visual aspect of the documents is usually analyzed in order to obtain a model of the semantic structures used in the document and the relations among them. In case of the HTML documents, the logical structure of the document can be discovered either by the analysis of the rendered document [3] or by the analysis of the document code [1, 7].

Finally, the information extraction from the logical document structure is mostly inspired by Shasha's work [5].

3 Visual Information Analysis

The visual information in a document is expressed by two basic components – the page layout and the visual attributes of the text. The layout of the page gives the user a basic idea of how the information is organized. Typically, the document is split into several parts, which are visually separated by different background color or by some kind of visual separators. The text attributes such as font size or color give more detailed information about the importance and relations among individual parts of the text in a particular area of the document.

We create separate models for both these components. The page layout model M_l describes the hierarchy of the visually separated areas in the documents whereas the text attributes model M_t is used for evaluation of the importance of individual parts of the document text.

3.1 Page layout model

In our approach we assume, that the visual areas of the document form a hierarchy where the root node corresponds to the whole document and other nodes correspond to the subareas that can be arbitrarily nested. Let's assign a unique identifier $v_i \in I$ to each area beginning with $v_0 = 0$ for the root area (the document) and assigning $v_i = v_{i-1} + 1$ to each new area encountered while reading the code. The page layout can be modeled as a hierarchy of the area identifiers.

Formally, the model of the page layout can be denoted as a graph:

$$M_l = (V_l, E_l) \tag{1}$$

where $V_l = \{0, 1, \ldots n-1\}$ is the set of all area identifiers and $(v_i, v_j) \in E_l$ iff v_i and v_j are the visual area identifiers and the area identified by v_j is contained in the area identified by v_i.

In our approach, we create the page layout model of an HTML document by locating the elements of the code that possibly form a visual area. However, any other approach can be used such as the whitespace-density graph analysis.

3.2 Text attribute model

An HTML document consists of the text content and the embedded HTML tags. Let's denote T_{html} a set of all possible HTML tags and S an infinite set of all possible text strings between each pair of subsequent tags in a document. Then, an HTML document D can be represented as a string of the form

$$D = T_1 s_1 T_2 s_2 T_3 s_3 \ldots T_n s_n T_{n+1} \tag{2}$$

where $s_i \in S$; $|s_i| > 0$ is a text string that does not contain any embedded HTML tags and n, $n \geq 0$ is the number of such strings in the document. T_j, $1 \leq j \leq n+1$ is a string of HTML tags

$$T_j = t_{j,1}t_{j,2}t_{j,3}\dots t_{j,m_j} \tag{3}$$

where $t_{j,k} \in T_{html}$ and $|T_j| \geq 1$.

Since the visual attributes of the text can only be affected by the HTML tags, each text string s_i has constant values of the attributes. Let's define the notion of *text element* as text string with visual attributes. For each text element, we define two numeric values that generally describe its visual apearance:

The *markedness* of the element determines how much the element is highlighted comparing to the remaining text of the document. For computing the markedness value x we use a simple heuristic: the font size is adequate to the markedness, the bold font, underlining and color-highlighting increases the markedness, whereas the strikeout denotes an insignificant text. The value x can be computed as

$$x = (F \cdot \Delta f + b + o + u + c) \cdot (1 - z) \tag{4}$$

where b, o, u and z have the value 1 when the text element is bold, oblique, underlined or striked-out respectively and 0 if they are not. Similarly, c indicates whether the text element has a color different from the document default. Δf is the difference $f - f_d$ where f is the font size of the element and f_d is the default font size for the document in points. The constant F defines the relation between the text size and its markedness. For $F > 4$ the element with greater font size is always more important than the element with lower font size. This corresponds to the usual interpretation.

The element *weight* expresses the relations of superiority and inferiority among the text elements. We suppose that there exists a hierarchy of headings and labels in the document. The most important heading has the highest weight and the normal text of the document has the lowest weight. For determining the element weight, following heuristics are used:

1. Labels and headings should be highlighted in the text. Therefore the weight of a label or heading is adequate to the markedness of the text element.
2. A bold, underlined or color-highlighted text element placed inside of the continuous text block is not a label. To be considered as a label, such a text element must be placed at the beginning of a text block.

3. Labels are often ended with a colon. Thus a trailing colon increases the weight of the element. An element that ends with a colon should have higher weight that possibly following bold, underlined or color highlighted elements.

Based on the above assumptions, the weight w of a text element is computed as

$$w = \Big[(F \cdot \Delta f) + (b + o + u + c) \cdot l + W \cdot p\Big] \cdot (1 - z) \tag{5}$$

where F, Δf, b, o, u, c and z have the same meaning as in (4), l and p have the value 1 when the element is preceded by the start of block or a line break and when the element text ends with a colon respectively and 0 in the opposite case. W is the weight of the trailing colon. The above specification in heuristic 3 is met for $W > 4$.

Each text element $e_i \in C$, where $C = S \times I \times I \times I$. As usual, we write e_i as

$$e_i = (s_i, v_i, x_i, w_i) \tag{6}$$

and we define

$$e_i < e_j, 1 \le i \le n-1, i+1 \le j \le n \tag{7}$$

where $s_i \in S$, $v_i, x_i, w_i \in I$. s_i is a text string that represents the content of the element, v_i is the visual area the element belongs to and x_i and w_i are the element markedness and weight.

The document with visual attributes can be modeled as a string of text elements of the form

$$M_t = e_1 e_2 e_3 \dots e_n \tag{8}$$

where $e_i = (s_i, v_i, x_i, w_i)$, $1 \le i \le n$ are the text elements. s_i corresponds to the appropriate text string in (2). v_i is determined during the tree of visual areas is being built.

4 Logical Document Structure

In our conception, the logical document structure is a hierarchy of all the text elements in the document where the elements at higher level of abstraction are described by the elements at finer level of abstraction.

The model of the logical document structure is based on the transformation of the visual information models. The basic hierarchy is determined by the hierarchy of the visual areas in the document. This hierarchy is further extended by the hierarchy of text elements within each area. Since the logical document structure is a tree of text elements, it can be denoted as

$$S = (V_S, E_S) \tag{9}$$

where $V_S = \{e_1, e_2, \ldots e_n\}$ is the ordered set of all the text elements in the document. The set of edges of the tree E_S is derived from the page layout model M_l (1) and the model of typographical attributes of the text M_t (8). This operation has two phases:

1. Creating the set of graph edges E_V such that $S_V = (V_S, E_V)$ is a tree of text elements and for any $e_i, e_j \in V_S$ e_i is an ancestor of e_j iff the corresponding visual area identifier v_i is an ancestor of v_j in the tree of visual areas M_l. We call S_V a *frame* of the logical document structure.
2. Creating E_S by copying all the edges (e_i, e_j) from E_V and replacing e_i by one of its descendants if needed, so that the element of a higher weight is always an ancestor of all the elements with a lower weight within the visual area.

The resulting model describes the document logical structure as expressed by the visual means.

5 Information Extraction from the Logical Structure

Once the logical structure is discovered, we can create a pattern of a subtree of the logical document structure that corresponds to the desired information. Let's consider a simple example mentioned in the introduction. From a personal page, we want to extract the name, department and the e-mail address of that person. In Fig. 1, we propose two possible variants of such a pattern that we will call *query trees*. For specifying the field formats and the possible labels, standard regular expressions have been used. The bold boxes correspond to the fields to be extracted. The first variant corresponds to the situation where the name of the person is used as a page title or a heading. The second variant corresponds to the situation where the name is presented on the same level as the remaining data fields. Now, the task is to locate the corresponding subtrees in the previously created logical structure tree.

For tree matching, we use a modified **pathfix** algorithm published by Shasha et al. [5].

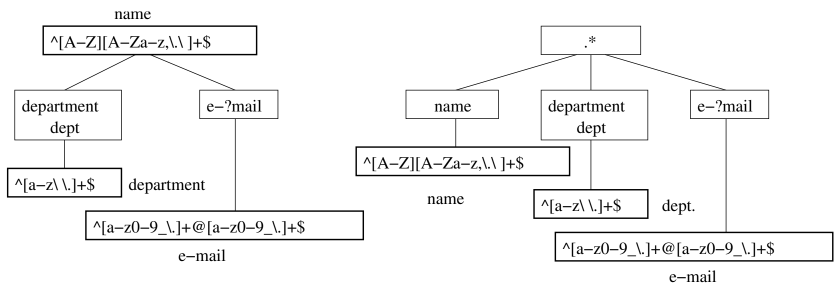

Fig. 1. The expected query tree with labels and the desired data (bold boxes) specified by regular expressions (variants A and B)

6 Experimental Method Evaluation

We have tested the method on the simple extraction task shown in Fig.1. As a data source we have used sets of staff personal pages from various universities. From each listed site, we have taken 30 random personal pages. The only input for the information extraction is the URI of the document and the query tree for the variant A or B. Table 1 shows values of *precision* and *recall*.

#	URL	Precision %	Recall %	Variant
1	`fit.vutbr.cz`	91	100	a
2	`mff.cuni.cz`	100	100	a
3	`is.muni.cz`	100	52	a
4	`mit.edu`	-	-	a
5	`cornell.edu`	100	100	a
6	`yale.edu`	100	100	a
7	`stanford.edu`	100	100	b
8	`harvard.edu`	100	100	b
9	`usc.edu`	-	-	b
10	`psu.edu`	100	100	b

Table 1. Sample extraction task results

This test shows that the results are independent on the HTML code (compare for example the sets #1, #2 and #5). However, there are following basic reasons that may cause the extraction process to fail: The extracted data is not labeled as expected (e.g. in the testing set 3 the e-mail addresses are not denoted by any label) or the data to be extracted is contained inside of larger text elements (insufficient visual formatting for the presented method).

7 Conclusions

In this paper, we have formally defined the visual information in HTML document and we have proposed the way of using it for information extraction.

The proposed method is suitable for extracting the data from structured data-intensive documents. In comparison to the methods based on the direct analysis of the HTML code, the main benefit of our method is the use of an abstract model of the visual information. Since the data is extracted from the logical structure model, the method is not dependent on the way how the visual appearance of the text elements has been achieved using HTML or other technologies.

On the other hand, the method is limited to the tasks, where the expected structure of the extracted data is fixed and the corresponding query tree can be exactly determined. This somewhat limits the amount of the possible tasks.

References

1. Chung, C. Y., Gertz, M., Sundaresan, N.: Reverse Engineering for Web Data: From Visual to Semantic Structures. In *18th International Conference on Data Engineering (ICDE 2002)*, IEEE Computer Society, 2002.
2. Freitag, D.: Information extraction from HTML: Application of a general learning approach. In *Proceedings of the Fifteenth Conference on Artificial Intelligence AAAI-98*. 1998
3. Gu, X.-D., Chen, J., Ma, W.-Y., Chen, G.-L.: Visual Based Content Understanding towards Web Adaptation. In *Proc. Adaptive Hypermedia and Adaptive Web-Based Systems*, Malaga, Spain, 2002, pp. 164-173
4. Hong, T.W., Clark, K.L.: Using Grammatical Inference to Automate Information Exraction from the Web. In *Principles of Data Mining and Knowledge Discovery.* 2001
5. Shasha, D., Wang, J.T.L., Shan, H., Zhang, K.: ATreeGrep: Approximate Searching in Unordered Trees. In *14th International Conference on Scientific and Statistical Database Management (SSDBM'02)* Edinburgh, Scotland, 2002
6. Summers, K.: Toward a taxonomy of logical document structures. In *Electronic Publishing and the Information Superhighway: Proceedings of the Dartmouth Institute for Advanced Graduate Studies (DAGS '95).* Boston, USA, 1995
7. Yang, Y., Zhang, H.: HTML Page Analysis Based on Visual Cues. In *Proc. of 6th International Conference on Document and Analysis*, Seattle, USA, 2001