Key Expression driven Record Mining for Event Calendar Search

Yeong Su Lee & Michaela Geierhos
Centrum für Informations- und Sprachverarbeitung
Ludwig-Maximilians-Universität München
{yeong|micha}@cis.uni-muenchen.de

Abstract
This paper presents an approach to extract data records from websites, particularly ones with event calendars. We therefore use language-specific key expressions and HTML patterns to recognize every single event given on the investigated web page. One of the most remarkable advantages of our method is that it does not require any additional classification steps based on machine learning algorithms or keyword extraction methods: it is a so-called one-step mining technique. Our experimental results obtained on German opera websites show excellent results in precision and recall. Furthermore, we could demonstrate that our proposed technique outperforms other data record mining applications run on event sites.

1 Introduction
There are numerous web sites providing large databases containing information such as yellow page listings or event calendars. A user typically accesses such a database over the Internet by using a web browser. The requested information is to be displayed in dynamically generated web pages by a script using a back-end database. During the last years, several parsing algorithms have been developed to automatically determine the information record boundaries and extract the corresponding data records.

Current approaches to data record mining [Arasu and Garcia-Molina, 2003; Crescenzi et al., 2001; Lerman et al., 2003] exploit the structured character of HTML documents. For this purpose, two ore more similar web pages have to be compared in order to extract the corresponding data records. These systems often expect preclassified web pages as input (cf. RoadRunner [Crescenzi et al., 2001]). Based on the fact that data records are dynamically generated from a back-end database, some applications like MDR [Liu et al., 2003] try to reconstruct the given web page benefiting from the regularities of HTML structure. Hereby, the main focus is to determine iterations of HTML tag sequences by using the DOM tree representation of a web page. Since most structured web data is arranged in rows and columns, many researchers have concentrated on table recognition for information extraction purposes [David W. Embley, 2006]. But the recent shift away from the HTML tag (table) to both cascading style sheets (CSS) and the (div) tag complicates the identification of data records.

Due to the loose strictness of HTML, others try to exploit the visual information provided by a web browser. Like [Cai et al., 2003] many researchers [Algur and Hiremath, 2006; Hiremath et al., 2005; Liu et al., 2006; Zhao et al., 2005] have proposed to use rendering techniques for data record mining. They apply these methods on the displayed query results in order to determine the record boundaries. Even flat and nested data records can be extracted by the VSAP technique [Hiremath et al., 2005] based on some heuristics [Algur and Hiremath, 2006]. Although rendering methods achieve good results, they have one big drawback: They require a web browser, that correctly displays the investigated web page, to determine some typical visual cues of a data region (e.g. size, background color, icons, font colors).

Regardless of the technique used, all methods have one point in common: Current approaches to data record mining disregard any language-specific information. We observed that existing data record mining techniques are not satisfactory enough for specialized search purposes limited to restricted domains. The success of event calendar search highly depends on language-specific trigger words indicating at least some date.

We therefore propose a novel and robust method for data record mining on demand. Browsing restricted domains allows us to define some key words, e.g. weekdays, nested in the data records of event sites. By means of limited vocabulary, we are able to analyze the document’s HTML structure and locate the corresponding data record boundaries. Our technique is quite robust in variability of the DOM, upgradeable and keeps data up-to-date. As our case study was limited to German opera websites, the investigated language was German. However, the developed technique is adaptive to non-German websites with slight language-specific modifications, and experimental results on real-life websites confirm the feasibility of the approach.

The paper is structured as follows. In the next section we introduce the concepts and terms used in the paper. In section 3, we present our data mining technique. Section 4 evaluates the proposed method. In section 5, we summarize our work and finally highlight future research directions.

2 Definition of terms
Terms that are used throughout this paper in a particular usage have to be clearly defined.

A large amount of information on the Web is contained in regularly structured objects, which we call data records.

1Visual Structure based Analysis of web Pages
2For research and test purposes the prototype of our system is available at http://www.cis.uni-muenchen.de/

yeong/EVENTSEARCH/eventsearch.html.
Such data records (...) often present the essential information of their host pages [Liu et al., 2003], e.g. event calendars.

Definition 1 (Event calendar record)
An event calendar record is primarily a data record which provides information on event details like event title, event location, event date, etc.

In order to distinguish event calendar records from ordinary data records hosting, for example, the search menu for event style, we have to define some vocabulary describing an event, so-called key expressions.

Definition 2 (Key Expression)
An instance of a feature set that classifies a data record and can be described by regular expressions or string variants is called key expression.

Every event calendar record contains useful information including weekday, date, time, and price (cf. Figure 1). Among these, we can easily identify date, time and weekday by using some regular expression and a list of weekdays together with their corresponding abbreviations.

date = [0-9]{2}\W[0-9]{2}\W[0-9]{4}\W
wdAbbrev = (Mo|Di|Mi|Do|Fr|Sa|So)
wd = ((Mon|Diens|Donners|Frei|Sams|Sonn)tag
  |Mittwoch|Sonnabend)

Other attributes like event title and price, and other additional information can be either difficult to recognize (e.g. title) or optional (e.g. price). Thus, we use date, time and weekday as key terms (a seed list) to search for an event calendar record. Having detected such a record, we extract all information bits found for the corresponding event.

Please note that the selection of a key expression highly depends on the record type. For example, within a shopping record, the key expression may be some price information, and within a computer description, it may be the CPU type.

3 The proposed technique
After retrieving a large website, each web page has to be classified into pages with event calendars or without, depending on its key expressions. If the page contains at least two or more key expressions, then the search for event calendar records will start. Otherwise, the page will be skipped. Thus, the classification of pages containing event calendar records can be performed without the help of machine learning algorithms or keyword extraction methods.

We now present the two steps of our approach:

1. First, we create the DOM tree of a selected web page in order to exploit its HTML structure (cf. section 3.1)
2. Secondly, we assume that there is only one smallest maximum data region for the event calendar records [Liu et al., 2006; Hiremath et al., 2005; Liu et al., 2003; Zhai and Liu, 2005] and it corresponds to only one HTML tag region. The smallest maximum data region can be determined by a top-down traversal of the tree using key expressions (cf. section 3.2). It is predictable that there must be two or more key expressions in one HTML tag region of the tree. Otherwise, this tag region will be cut off from the DOM tree.

3.1 Exploiting the structure of event calendar records within the DOM tree
Each website has its own distinct method of presenting information. Therefore, the high variability observed in HTML structure should be taken into account. However, the number of possible tag combinations which can be considered for event calendar records is very limited.

The following types of event calendar records according to their tree structure have been registered and were classified as follows:

1. One single record under one node
2. All records under one node – each record consists of a set of children nodes (cf. Figure 2)

In the first case, we act on the assumption that each data record represented by one HTML tag region has no siblings. If this tag region contains some key expression (e.g. weekday) and other event-related information, it will be selected as event calendar record.

In contrast to the first type, all event calendar records are siblings (cf. Figure 2) belonging to the same parent node.
We can thereby distinguish between three structure types of data records (HTML tag regions) depending on the co-occurrence of tag attributes and values.

(a) repetition of an HTML tag, e.g. \texttt{(div)} with non-recurring attributes across all data records, including their text values.

(b) repetition of an HTML tag, e.g. \texttt{(div)}, with recurring attributes across all data records and sometimes incomplete attribute-value-pairs (cf. Figure 3),

(c) missing both tag attribute and value.

Among these, case (a) is really rare and (c) can sometimes happen, but in practice, case (b) occurs quite frequently.

In Figure 2, we showed that the key expression is inherited to only one HTML tag node (\texttt{html} \rightarrow \texttt{body} \rightarrow the 5\textsuperscript{th} \texttt{(div)} \rightarrow the 4\textsuperscript{th} \texttt{(div)}), and all records are the children of this one single node (cf. \texttt{div class=\textasciitilde\texttt{content}}") in Figure 3). When we zoom in and look at the record structure in detail, each record is composed of six \texttt{(div)} tags and its corresponding attributes: "\texttt{kalendariumtag}"", "\texttt{kalendariumdatum}"", "\texttt{kalendariumuezeit}"", "\texttt{kalendariummitte}"", "\texttt{kalendariumpreise}" and "\texttt{kalendariumlinie}". As shown in Figure 3, the text values (\texttt{c}) are missing for the first two mentioned HTML tag attributes in the one record, but are filled with \texttt{#PCDATA "A1"} and \texttt{#PCDATA "B1"} in the preceding record. This means that the text values of the first record are also valid for the following record. In our case, the opera \textit{Don Giovanni} takes place on the same day (\texttt{kalendariumtag=Sa, kalendariumdatum=14}) as \textit{Aida} (\texttt{kalendariumtag=\epsilon, kalendariumdatum=\epsilon}). We therefore resolve such co-references by linking the text values of the same attributes in successive records.

Assuming that the two tag attributes displayed in gray are also missing, Figure 3 would be an example for (\texttt{c}).

But one problem that still remains to be solved is how to decide where the record starts and where it ends: The boundary between records can be determined by comparing the bordering tag attributes. Based on the assumption that the key expression, e.g. \textit{weekday}, is placed in first position (cf. \texttt{#PCDATA "A1"} in Figure 3), then we have two possibilities: We can go forward or backward to recognize the record boundaries. If we move forward, the same tag attribute will recur after six steps. That way, we learn that one record consists of six tag attributes. However, we do not know yet where the record begins. In order to solve this problem, we go back until we find a tag attribute totally different from the six common attributes in Figure 3. Now we can initialize the starting points for all records embracing six tag attributes each.

In all cases, we try to correctly determine the smallest maximum data region.

### 3.2 Decision of smallest maximum data region

After classifying an HTML document as event calendar page, we have to detect the smallest maximum data region containing some key expression within the DOM tree. Assuming that the smallest maximum data region only consists of event calendar records, every record must have its own parent node.

```plaintext
1 sub scanSmallestMaxDataRegion {  
2     node = shift;  
3     for each child (node->contentList)  
4         if (child has a set of nodes)  
5             if (child->asHTML() matches keyExp)  
6                 then smallestMaxDataNode = child;  
7                     scanSmallestMaxDataRegion(child);  
8                 else  
9                     scanSmallestMaxDataRegion(child);  
10                 endif  
11             else  
12                 return 0;  
13             endif  
14         endif  
15     return smallestMaxDataNode;  
16 }
```

In Figure 4 is described how to determine the smallest maximum data region. In short, we launch a top-down-traversal of the DOM tree starting at the root node. We then search the content of the child nodes for key expressions in order to detect the regularities in their occurrences. That way, we measure the maximum distance of a re-appearing type of key expression, e.g. \textit{weekday}, and determine the
corresponding subtree within the HTML structure. Our understanding of data region is thereby the same as in [Hiremath et al., 2005]. Once, the smallest maximum data region is located, each record must be mined by using some key expression.

4 Experimental evaluation

To evaluate the quality of the proposed record mining technique from arbitrary websites, we concentrate our case study on websites of German opera houses. Our test set consists of eleven event calendar pages randomly retrieved from websites of opera houses dated on April 14, 2009.

<table>
<thead>
<tr>
<th>URL</th>
<th>Record</th>
<th>KEDR1</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>bayerische.staatsoper.de</td>
<td>21</td>
<td>20</td>
<td>95.24%</td>
</tr>
<tr>
<td>staatsoper-berlin.org</td>
<td>33</td>
<td>33</td>
<td>100.00%</td>
</tr>
<tr>
<td>theater-chemnitz.de</td>
<td>10</td>
<td>10</td>
<td>100.00%</td>
</tr>
<tr>
<td>oper-frankfurt.de</td>
<td>26</td>
<td>26</td>
<td>100.00%</td>
</tr>
<tr>
<td>oper-halle.de</td>
<td>24</td>
<td>24</td>
<td>100.00%</td>
</tr>
<tr>
<td>hamburgische-staatsoper.de</td>
<td>17</td>
<td>17</td>
<td>100.00%</td>
</tr>
<tr>
<td>oper-hannover.de</td>
<td>26</td>
<td>26</td>
<td>100.00%</td>
</tr>
<tr>
<td>oper-leipzig.de</td>
<td>27</td>
<td>27</td>
<td>100.00%</td>
</tr>
<tr>
<td>rheinoper.de</td>
<td>12</td>
<td>12</td>
<td>100.00%</td>
</tr>
<tr>
<td>semperoper.de</td>
<td>37</td>
<td>37</td>
<td>100.00%</td>
</tr>
<tr>
<td>staatstheater.stuttgart.de</td>
<td>37</td>
<td>37</td>
<td>100.00%</td>
</tr>
<tr>
<td><strong>On average</strong></td>
<td></td>
<td></td>
<td><strong>99.56%</strong></td>
</tr>
</tbody>
</table>

 Needless to say, the evaluation results displayed in Table 1 show excellent results of recall (99.56%) and we constantly obtained a precision of 100%. The reason for this lack of recall is due to the non-recognition of the time attribute within a data record. As mentioned in section 2, every event calendar record contains weekday, date, time, title and price information. For a full recognition of a record, at least weekday and time have to be correctly determined. Assuming that one aspect is missing, we count this record as false negative. For lack of specification, our algorithm could not identify the time for one stage performance in April’s event calendar of Bayerische Staatsoper (95.24%).

One presumption is that our key expression driven record mining technique expects a valid HTML page for the DOM tree construction. If the tree cannot be built up, we will use some open source tools, e.g. tidy, for correction purposes. We must admit that our approach is not able to reconstruct the DOM tree of web pages with no closing HTML tags. Running other comparable data record mining applications like MDR [Liu et al., 2003] on our test web pages does not produce any noteworthy results. The event calendar records cannot be either located or correctly assigned to the corresponding data regions. Therefore, recall and precision are vanishing small.

5 Conclusion and future work

Since current approaches to data record mining have disregarded any language-specific information and only exploited the structured character of HTML, we combine both: HTML patterns with predefined key expressions.

Our future work will concentrate on both automated key expression learning and substructure analysis of data records. By measuring the similarity of content strings or tag regions, we will figure out the best candidates for domain-specific key expressions. Moreover, it seems essential to validate this approach on a much larger test set demonstrating that web pages of that type show little variability. Besides, it could be interesting to test this technique on other websites with event information (e.g. sports).

References


