

Using the Structure of Web Sites for Automatic Segmentation of Tables

by Kiristina Lerman, Lise Getoor, Steven Minton and Craig Knoblock, 2004

1. Focus

The paper describes two approaches to **unsupervised extraction** of structured data from “*hidden web*” - sites. These pages are dynamically generated as results of user-queries for telephone numbers, electronic parts, books etc. A main aspect of the considered work is the aim that the segmentation process should work **domain independent** and **fully automatic**. Fulfilling these demands, the extraction method can be applied to various web sites, adapt easily to site-changes and perform well on the increasing amount of hidden web-content without consuming much human resources for generation and update of training examples.

2. Problems of layout-based segmentation

The task of table- or record extraction is not a trivial one, because a great **variability of table-styles and layout methods** are used to place tables on screen. Compared to plain text, where the use of white spaces is common to all table layouts, in HTML there are lots of possibilities like non-standard table tags and separators to lay out the table-data. The authors state that using table tags (`<td>`,...) as an indicator for records will rarely give good result, as these **tags are often used for multi-column-text or layout** of images. Text separators (`
`, “~”,...) are used for the separation of fields as well as items. *Pat-Trees* are used to identify repeated HTML-tag-sequences, but they suffer from **limited utility** when applied on more complicated sites. *Web-wrappers* that learn the structure of a site usually are domain specific heuristics and rely on **training examples**. Grammar-based algorithms like *RoadRunner* cannot handle **disjunctions** which appear when layout attributes change dynamically.

3. Assumption and Methods

A possible strategy to overcome many of these problems is the use of a structure that is common to many “hidden-web”-sites generated by web-queries:

- *) a HTML-Form is presented to the user that allows the input of a query.
- *) an overview of results is automatically generated as answer to the query. this overview is called **list-page** and consists of a description and a link to the **detail-page** referring to that item.
- *) these detail pages are automatically generated when a link from the list-page is chosen. The detail-page presents more information for the corresponding item. So there are two views of the same item (record). The **redundant information in the content** of detail-pages could point to a possible record segmentation of items on the list page.

3.1 Page Sources & Tokenization

The first step in the segmentation process is to download a set of page sources (list- and according detail pages). The sources are tokenized, using the following **tokens types** suggested by the authors: “HTML”, “punctuation”, “alphabetic”, “numeric”, “capitalized”, “lowercased”. Other type like “bold” or “italic” could be used, too. The tokens serve as input to the segmentation algorithms.

3.2 Templates and Slots

As list- and detail-pages are generated automatically, there exists an **invariant part** of these pages, containing headers and footers, advertisements, summaries, copyright info, navigation aids etc. This part of the page is called **template**. There are templates for the whole page and table-templates, which contain table header. Templates can be recognized by comparing two or more pages of the same type.

Varying data or repeating patterns (that are considered to be elements of a table) are not part of the template. These parts of the page are called **slots**. The slot with the largest number of text tokens in it is considered to contain the interesting information.

3.3 Extraction

For a given slot, a contiguous sequence of text tokens is extracted, until the point where the next HTML or separator token appears. This leads to n **extracts of visible strings**: $E = \{E_1, E_2, \dots, E_n\}$. We want to assign these extracts E_i to records. Same extract values can appear on multiple detail pages (for example same name or telephone number for different people).

The detail pages found on the list page are called $\{r_1, r_2, \dots, r_k\}$, D_i are the detail pages on which a specific extraction value E_i appears. If an extract appears in every detail page, it will be discarded. If an extract does not appear on list and detail page, it will be discarded, too. -> only extracts that appear on both are considered.

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
	John Smith	221 Wa ington...	New Holland...	(740) 335-5555	John Smith	221R Wa shington...	Wash ington...	(740) 335-5555	George W. Smith	Findlay, OH...	(419) 423-1212
D_i	$r1, r2$	$r1$	$r1$	$r1, r2$	$r1, r2$	$r2$	$r2$	$r1, r2$	$r3$	$r3$	$r3$

Fig1: example extraction from a white pages - query (superpages.com)

4. Segmentation Approach I: Constraint Satisfaction Problem (CSP)

CSPs are stated as logical expressions (or constraints) over a set of variables. The values of variables are 0 or 1 (pseudo-boolean representation). Solving the CSP gives an assignment to the variables such that all constraints are satisfied at the same time. Inequalities are allowed, their use generates optimization problems. To formulate the CSP, we need an assignment variable x_{ij} which is 1 when extract E_i is assigned to record r_j . Following rules can be written as constraints:

Uniqueness constraint:

Every extract belongs to exactly one record r_j : $\sum_j x_{ij} = 1$

Consecutive constraint:

Only contiguous blocks of extracts can be assigned to the same record:

$$x_{kj} + x_{ij} \leq 1 \text{ when } x_{nj} = 0, k < n < i$$

Constraints derived from structural assumptions:

If extract E_i was not observed on detail page r_j then x_{ij} is 0

If extract E_i was observed on detail page r_j then x_{ij} is 1 or 0

Position constraints:

Detail pages provide information about the position of an extract.

If two extracts appear on the same position, they have to be assigned to

$$\text{different records: } \text{pos}(i) = \text{pos}(k) \rightarrow x_{ij} + x_{kj} = 1$$

The CSP can be solved using programs like **WSAT(OIP)**, which is freeware.

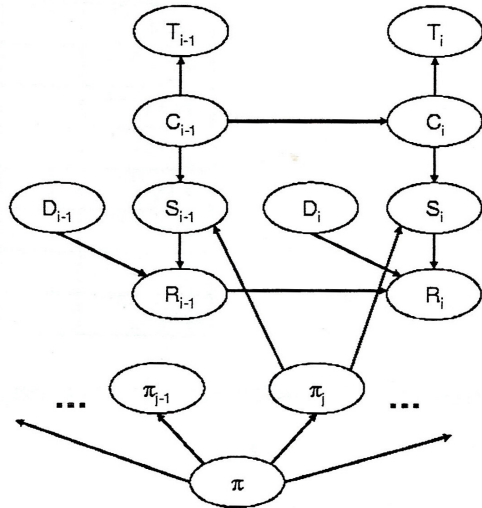
The solution is an assignment of extracts to records.

	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}	E_{11}
	John Smith	221 Wa ington...	New Holland...	(740) 335-5555	John Smith	221R Wa shington...	Wash ington...	(740) 335-5555	George W. Smith	Findlay, OH...	(419) 423-1212
$r1$	1	1	1	1							
$r2$					1	1	1	1			
$r3$									1	1	1

Fig2: solved CSP yields record segmentation for the above example

5. Segmentation Approach II: Probabilistic Model (HMM)

This method tries to frame the record segmentation and extraction task as a probabilistic inference problem. A **Hidden Markov Model** (HMM) is used for this purpose. HMMs can be used to learn parameters of a system with a number of **observable states** that depend on a so-called **“hidden-state”**. For the given task of record segmentation, the “hidden state” is **factorized** to reduce its possibilities and to find a better representation for the dependencies:



Variables:

- $T = \{T_1, \dots, T_n\}$ token-types of extract E_i
- $D = \{D_1, \dots, D_n\}$ detail pages where E_i occurred
- $R = \{R_1, \dots, R_n\}$ record number of the extract (to be found)
- $C = \{C_1, \dots, C_n\}$ column label of the extract (to be found)
- $S = \{S_1, \dots, S_n\}$ true if E_i is the start of a new record, false otherwise

π_i represents the table period (number of columns). It can vary from record to record (missing columns), but will most likely be the total number of columns. π is a global parameter that adds a hierarchical **structure** to the model and makes it more traceable.

Dependencies:

- $P(T_i|C_i)$: token type depends on column label. (e.g. the name is written capitalized)
- $P(C_i|C_{i-1})$: column label depends on label of the previous column. (e.g. the address follows the name-field)
- $P(S_i|C_i)$: start of a new record depends on column label. A deterministic assumption can be made here: the first column is most important and never missing:
- $P(R_i|R_{i-1}, D_i, S_i)$: record number for extract E_i depends on record number of the previous extract, on the start of a new record and on the detail pages on which E_i was observed. This is in general a deterministic relationship:
if S_i is false, then $P(R_i = R_{i-1}) = 1.0$, if S_i is true, then $P(R_i = R_{i-1} + 1) = 1.0$

To find further constraints, information from the detail pages or initial probabilities for token-types can help to make initial assignments (**bootstrapping**). The task of record segmentation boils down to finding values for the unobserved R and C variables.

Expectation maximization (**EM**) and the Forward-Backward Algorithm (**Baum-Welch**) can be used to implement the model. The algorithm computes most likely assignment for R and C .

6. Results

The two approaches have been tested with queries from 12 different Web sites. Some used grid-like tables, with or without borders, others were more free-form, consisted of blocks etc. Entries could be numbered or unnumbered. The results of the algorithms were classified as Cor (correctly segmented), InCor (incorrectly segmented), FN (unsegmented records) and FP (non records)

Precision and **Recall** were computed:

$$P = \text{Cor} / (\text{Cor} + \text{Incor} + \text{FP}) \quad (0.85 \text{ for CSP approach, } 0.74 \text{ for probabilistic approach})$$

$$R = \text{Cor} / (\text{Cor} + \text{FN}) \quad (0.84 \text{ for CSP approach, } 0.99 \text{ for probabilistic approach})$$

7. Discussion

A significant amount of false classifications was influenced by the weak template-finding algorithm (5 cases). The CSP approach was very reliable on clean data, but sensitive to errors and inconsistencies in the data source. The HMM tolerates inconsistencies better and is more expressive in sense of a column assignment.

Content specific algorithms for web-extraction are not as widely used as layout based techniques, but they show advantages in speed (there is less data in plain text than in the HTML-layout) and in utility (unsupervised, widely independent from layout-implementations). On the other hand, the utility of the described techniques is restricted by the fact that they rely on special relations between list- and detail-pages: If given attributes do not appear on both list- and detail page or are written in a different way, the segmentation algorithms will not be able to segment the data correctly.

Literature:

- K.Lerman, L.Getoor, S.Minton, C.Knoblock: Using the Structure of Web Sites for Automatic Table Segmentation, *Proceedings of the 2004 ACM SIGMOD international conference on Management of data, Paris, France*
- L.R.Rabiner: A tutorial to hidden markov models and selected application in speech recognition. In *Readings in Speech Recognition*
- J.P. Walser: WSAT(OIP) - Local Search for Over-constrained Integer Programs, Universitaet des Saarlandes, Programming Systems Lab, 1997