

Extracting Attributes and Their Values from Web Pages

Minoru YOSHIDA^{†, ‡}

[†] Department of Information Science, Graduate School of Science, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan

[‡] CREST, JST (Japan Science and Technology Corporation)
4-1-8 Kawaguchi Hon-cho, Kawaguchi-shi, Saitama, 332-0012, Japan
mino@is.s.u-tokyo.ac.jp

Abstract

We propose a method for extracting *attributes* and their *values* from Web pages. Our method makes use of word distributions estimated from plain Web pages. The key idea is to estimate word distribution by consulting ontologies built from HTML tables. In a series of experiments, we show that estimated word distributions are useful for extracting attributes and their values in various kinds of HTML representations other than tables.

1 Introduction

This paper describes a method for extracting *attributes* and their *values* from Web pages. In Web pages, although a sentence is the standard means to express information, many other styles of representation, such as tables or lists, are also used. In particular, important data are often expressed in the form of *attributes* and their *values*. About-me pages provide profiles of people by listing their names, sexes, addresses, etc.; and PC catalogue pages describe PCs by listing CPUs, memory sizes, etc. Extracting attributes and their values are therefore useful for various applications, such as the automatic summarization of Web pages.

This study is aimed at extracting attributes and their values represented as *non-sentential blocks*, which are the parts of Web pages containing no sentences. Figure 1 shows example pages consisting of non-sentential blocks. For example, attribute-value

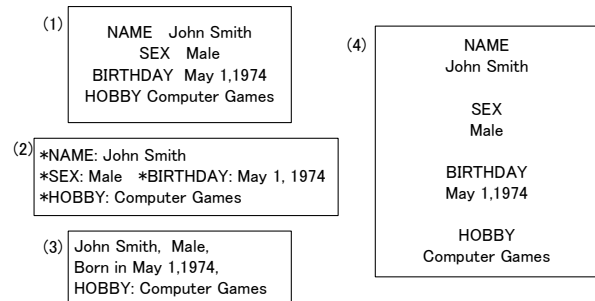


Figure 1: Various formats for the same content. There are a variety of attribute indicators. In some cases attributes are entirely omitted.

pairs (“NAME”, “John Smith”) or (“SEX”, “Male”), etc., are extracted from the page (1) in Figure 1.

Our method is distinguished by the following two features. First, it needs no manual interventions such as labeling of training samples or constructing of extraction patterns. Our idea is to consult *ontologies extracted from HTML tables*, to estimate the probability of each word occurring on Web pages. In this study we assume that an ontology is a description of some class of objects described by attribute-value pairs as exemplified in Figure 2¹. In our previous work we demonstrated a method to extract ontologies from tables with no labeled training samples (Yoshida et al., 2001). Ontologies used in our system are extracted by this method. Second, the method relies on only distributions of words, not HTML tags or other expression forms, to recognize

¹Notice that the usage of the term “ontologies” in this paper is somewhat different from the standard one (which is mostly used to represent “concept hierarchies”).

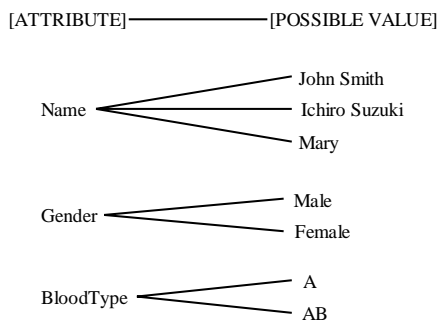


Figure 2: An example ontology of human. Each attribute or value is represented by a sequence of strings expressing it.

attributes and values on Web pages. Our algorithm therefore can deal with many types of presentational styles such as shown in Figure 1.

There have been some researches on extracting data from Web pages with no extraction patterns constructed by hand. Kushmerick et al. (1997) proposed *wrapper induction*, which automatically learned rules for extracting data from Web pages. Some approaches to this task were also proposed (Kushmerick et al., 1997; Hsu and Dung, 1998; Muslea et al., 1999). Freitag (1998) showed a method for extracting information from Web pages by using relational learning with features making use of HTML tags. However, all of these approaches requires that training data labeled with the target fields be extracted by hand.

The remainder of this paper is organized as follows. In Section 2, we give the term definitions. In Section 3, we explain our system in detail. Section 4 shows the results of some preliminary experiments and Section 5 concludes this paper and shows the future direction of this work.

2 Assumptions and Definitions

In this section we first give some assumptions on Web pages. After that, we define the terms used in the remainder of this paper.

2.1 Web Pages as Block Sequences

Web pages consist of not only sentences, but also *non-sentential blocks* such as tables and lists, as shown in Figure 1. Although there are various types of expressions even for the same content, they can be

treated uniformly in the form of a *sequence of blocks* separated by HTML tags or other special characters. Block sequences can be seen as the variant of text documents (i.e., a sequence of sentences) because blocks and sentences are the same in that both of them can be seen as sequences of words. Therefore, traditional statistical models for text documents could also be applied to Web documents containing non-sentential blocks. Among existing statistical models, we chose Hidden Markov Models (HMMs) because neighboring blocks are often closely related like *SEX* and *Male* in the example pages in Figure 1.

2.2 Term Definitions

In the following we give definitions of the terms used in this paper.

- A *page* is a sequence of *page fragments*, each of which is either a *block* or a *separator*.
- A block *b* is a sequence of words.
- A separator is a sequence of HTML tags or *special characters*. The special characters are characters which tend to be used as boundaries of blocks. They are defined a priori².
- An *ontology* is a sequence $\langle (A_1, V_1), (A_2, V_2), \dots, (A_m, V_m) \rangle$, where A_i and V_i correspond to the i th attribute in the ontology and its value, respectively. A_i is a sequence of the strings used in expressing the i th attribute and V_i is that used in expressing its value.
- A *role* is a pair (l, i) , where $l \in \{att, val\}$ and $i \in \{1, 2, \dots, m\}$. l , or a *label*, denotes whether a block represents an attribute or a value, and i , or an *index*, denotes the attribute's (or value's) number in the ontology.
- A *state* is defined for each block and has a role as its value. We denote the label of the state s by $l(s)$ and the index by $i(s)$.

²Currently we have 23 special characters, including “:”, “#” and “=”.

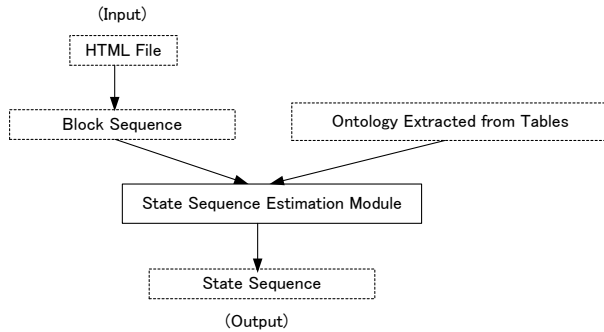


Figure 3: System overview. The SSEM, the main module of the system, uses the ontology extracted from tables.

3 System Overview

In this section we introduce our system and its algorithm in detail. Figure 3 shows the overall workflow of our system. A Web page given as an input to our system is first decomposed into a sequence of blocks bounded by separators. The State Sequence Estimation Module (SSEM) determines a sequence of states for the block sequence, by using an ontology extracted from HTML tables.

There are the following problems to be solved through this process.

Filtering out useless parts of pages There are many blocks that do not include any attribute or value. Filtering out such useless blocks is an important task because naive estimation of probabilities results in assigning some role even to such blocks. To solve this problem, we define additional roles called *a sentence role* and *a rare role*. This will be described in Section 3.3.

Combining separated but semantically-continuing parts Sometimes one role crosses over two or more neighboring blocks because separators are sometimes used only for the purpose of adjusting the appearance of a page. For example, the page (1) in Figure 1 uses a space as a separator, which separates values such as *John Smith* into more than one block. We introduce a model of *boundaries of roles* to solve this problem in Section 3.4.

In the remainder of this section, we briefly explain ontology extraction from tables first. After that, a

method to estimate a sequence of states is described.

3.1 Ontology Extracted from Tables

Our system uses an ontology extracted from HTML tables. There has been some research on the task of extracting ontologies from HTML tables on the WWW (Chen et al., 2000; Yoshida et al., 2001). In this research, we used the algorithm described in our previous paper (Yoshida et al., 2001) to extract ontologies. Each ontology output by this algorithm has a form defined in the previous section: a sequence of pairs of sequences of strings. We made a sequence of words from each sequence of strings in an ontology by decomposing each string into words by using a Japanese morphological analyzer JUMAN (Kurohashi and Nagao, 1998). We use a unigram model for each sequence of words. From each sequence of words, a frequency of a word-role pair $C(w, r)$ is enumerated as the number of times that w appears in r . This $C(w, r)$ is the base of calculation of probabilities in the SSEM.

3.2 State Sequence Estimation Module

Given a sequence of blocks $\mathcal{B} = \langle b_1, b_2, \dots, b_n \rangle$, the State Sequence Estimation Module (SSEM) estimates the most probable sequence of states $\mathcal{S} = \langle s_1, s_2, \dots, s_n \rangle$. Here, s_i is a state given to the block b_i .

The SSEM estimates the \mathcal{S} so that $P(\mathcal{S}|\mathcal{B})$ takes the highest value. In other words, \mathcal{S} is estimated according to the following formula.

$$\begin{aligned}
 \hat{\mathcal{S}} &= \arg \max_{\mathcal{S}} P(\mathcal{S}|\mathcal{B}) \\
 &= \arg \max_{\mathcal{S}} \frac{P(\mathcal{S}, \mathcal{B})}{P(\mathcal{B})} \\
 &= \arg \max_{\mathcal{S}} P(\mathcal{S}, \mathcal{B})
 \end{aligned}$$

We use an HMM as a model for \mathcal{B} . In this model, $P(\mathcal{S}, \mathcal{B})$ is calculated as follows:

$$P(\mathcal{S}, \mathcal{B}) \approx \pi(s_1)q(b_1|s_1) \prod_{i=2}^n p(s_i|s_{i-1})q(b_i|s_i) \quad (1)$$

where $\pi(s)$ is the probability of s appearing as the first state, $p(s'|s)$ is the probability of transitions between s and s' , and $q(b|s)$ is the output probability of b from the state s . Based on this formula, the most

probable sequence \mathcal{S} is calculated by the standard Viterbi algorithm (Forney, 1973).

In the following sections, we explain how to estimate values of $q(b|s)$, $p(s_i|s_{i-1})$ and $p(s)$, all of which are needed to calculate the above probability.

3.3 Estimation of $q(b|s)$

As defined above, a block is a sequence of words. We use unigram models for blocks. In these models, the probability $q(b|s)$ is calculated as $\prod_{i=1}^{|b|} q(w_i|s)$ where $b = \langle w_1, \dots, w_{|b|} \rangle$. Each $q(w|s)$ is estimated as

$$\frac{C(w, s)}{\sum_{x \in W} C(x, s)}$$

where W is a set of all the words appearing in the ontology. The main problem is that there are many pairs with zero frequencies, that is, the pairs (w, s) such that $C(w, s) = 0$. It is rare that, for some state s , all the words in a block have a non-zero value of $C(w, s)$. For this reason, it will be problematic to use just this definition of $q(w|s)$ because most probabilities become zero.

Currently, we use the Good-Turing Estimation (Good, 1953) to solve this problem. In the Good-Turing Estimation, the frequency t is adjusted as $t^* = (t + 1)N_{t+1}/N_t$ where N_t is the number of the kinds of pairs (w, s) which occurred just t times in the training data. In practice, this adjusted frequency is used only for the t such that $t \leq k$. (k is a threshold and currently set to 5.)

Another important point is that our algorithm makes the following special roles to filter out the blocks that do not represent any attribute or value.

Rare role The algorithm selects “top N attributes” for each ontology and classifies other attributes and their values into a *rare role*³. All roles classified as a rare role are treated as the same role. Top n attributes are selected according to their frequencies (i.e., $|A_i|$). For example, if $N = 10$, totally 20 roles (10 attributes and their values) are selected. Because rare roles are made from various kinds of roles, their word distributions are averaged and have no special characteristics. This gathering of low-frequency roles makes it possible to filter out a block which has no peculiar distribution of words because such a

³We call other (top) roles *non-rare* roles. Notice that it does not include the sentence role described below.

block is likely to be given with the rare roles rather than any other particular role.

Sentence role If a string in an ontology has at least one period, question mark or exclamation mark, words in the string are classified into a *sentence role*. A block with the words likely to appear in sentences, such as conjunctions or auxiliary verbs, tend to be classified into this role. It contributes to the filtering out of sentences which do not represent any role.

3.4 Estimation of $p(s_i|s_{i-1})$ and $\pi(s)$

We estimate $\pi(s)$ based on relative frequencies of s in ontologies.

$$\hat{\pi}(s) = r(s)$$

where

$$r(s) = \frac{\sum_w C(w, s)}{\sum_{s'} \sum_w C(w, s')}.$$

$p(s_i|s_{i-1})$ is estimated according to the following heuristics.

- An attribute must be followed by its value.
- A state is likely to be followed by the same state.

The former heuristic is expressed by the following constraint.

Constraint-1 If $s_i \neq s_{i-1}$ and $l(s_{i-1}) = att$, s_i must be $(val, i(s_{i-1}))$.

The latter heuristic reflects a problem of *boundaries of roles*. Although a block is a unit of roles in Web pages, it is not ensured that one role corresponds to just one block. Rather, two or more neighboring blocks often play the same role (See Figure 4). The algorithm needs boundaries of roles besides those of page blocks to recognize attributes and their values accurately.

To model boundaries of roles, let us encode them by using a sequence of bits. In this sequence, a bit in the i th position indicates whether a block is the end of a role (1), or not (0) (See Figure 4). We assume that all patterns of boundaries (which are equivalent to all patterns of the bit sequences) have the same probability. The probability of every bit sequence d is therefore given as $\frac{1}{2^{|d|}}$ where $|d|$ is the length of d .

| | | | | | | | | |
|-------------|------|----------|-------|---------|-------|-----------|----------|-------|
| Blocks | NAME | John | Smith | AGE | 23 | HOBBY | Computer | Games |
| Roles | NAME | val:NAME | AGE | val:AGE | HOBBY | val:HOBBY | | |
| Boundaries: | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |

Figure 4: Blocks, Roles, and Boundaries as a bit sequence

In this model, the probability $P(\mathcal{S}, \mathcal{B})$ is revised as follows.

$$\begin{aligned}
P(\mathcal{S}, \mathcal{B}) &\approx \sum_d P(d) \prod_{i=1}^n p'(s_i | s_{i-1}, d) q(b_i | s_i) \\
&\approx \sum_d \frac{1}{2^{|d|}} \prod_{i=1}^n p'(s_i | s_{i-1}, d_{i-1}) q(b_i | s_i) \\
&= \sum_d \prod_{i=1}^n \frac{1}{2} p'(s_i | s_{i-1}, d_{i-1}) q(b_i | s_i) \\
&= \prod_{i=1}^n \sum_{d_{i-1}} \left\{ \frac{1}{2} p'(s_i | s_{i-1}, d_{i-1}) \right\} q(b_i | s_i) \\
&= \prod_{i=1}^n \left\{ \frac{1}{2} p'(s_i | s_{i-1}, d_{i-1} = 0) \right. \\
&\quad \left. + \frac{1}{2} p'(s_i | s_{i-1}, d_{i-1} = 1) \right\} q(b_i | s_i)
\end{aligned}$$

where $p'(s_1 | s_0, d_0) = \pi(s_1)$, and $p'(s' | s, d)$ is the transition probability between s and s' assuming that a bit sequence is d . The last expression has the same form as that derived from equation 1 by replacing the $p(s_i | s_{i-1})$ with the expression

$$\frac{1}{2} p'(s_i | s_{i-1}, d_{i-1} = 0) + \frac{1}{2} p'(s_i | s_{i-1}, d_{i-1} = 1).$$

So we use this expression as the revised definition of $p(s_i | s_{i-1})$.

We set the following constraint to $p'(s_i | s_{i-1}, d_{i-1})$ corresponding to the second heuristic.

Constraint-2 $p'(s_i | s_{i-1}, d_{i-1})$ must be 0 if $d_{i-1} = 0$ and $s_i \neq s_{i-1}$, or if $d_{i-1} = 1$ and $s_i = s_{i-1}$.

It leads to the formula $\hat{p}(s_i | s_{i-1}) = \frac{1}{2}$ for $s_i = s_{i-1}$. (It corresponds to the case where $d_{i-1} = 0$.) The remaining probabilities (for the case when $d_{i-1} = 1$) are distributed among other values of s_i according to the relative frequencies $r(s_i)$. As the consequence, we derive the estimation

$$\hat{p}(s_i | s_{i-1}) = \frac{1}{2} \cdot \frac{r(s_i)}{\sum_{s \neq s_{i-1}} r(s)}$$

when $s_i \neq s_{i-1}$ and $l(s_{i-1}) = val$, and

$$\begin{cases} \hat{p}(s_i | s_{i-1}) = \frac{1}{2} & \text{if } i(s_i) = i(s_{i-1}), l(s_i) = val \\ \hat{p}(s_i | s_{i-1}) = 0 & \text{otherwise} \end{cases}$$

when $l(s_{i-1}) = att$ (by the Constraint-1).

4 Preliminary Experiments

To evaluate our system, we gathered three sets of Web pages⁴: 20 about-me pages, 20 PC specification pages and 20 company profile pages. These pages did not include any `<table>` tag. We picked up one corresponding ontology for each page set (for example, an ontology describing human entities for about-me page set), by hand, among the ones extracted from tables⁵. The number of non-rare roles for each ontology was set to 20. All the blocks in those pages that have non-rare roles were extracted and labeled with the correct roles by hand. Performance was evaluated by comparing the roles output by our algorithm with the ones labeled by hand. Precision is given by N_c/N_m and recall is given by N_c/N_h where N_m is the number of roles output by the algorithm, N_h is the number of roles given by hand and N_c is the number of times roles output by hand and those output by the algorithm were the same. F-measure was calculated as $F = (2 \cdot recall \cdot precision) / (recall + precision)$. We also evaluated the accuracy when a Naive Bayes Classifier was used for each block, where the state for each block $b = \langle w_1, \dots, w_{|b|} \rangle$ was estimated as

$$\arg \max_s r(s) \prod_{i=1}^{|b|} q(w_i | s).$$

The result for Naive Bayes Classifier can be seen as the result when the dependencies between states was not used in the HMM.

Table 1 shows the result. In general, recall was improved by the use of HMMs in comparison with the Naive Bayes Classifier, while precision remained the same or decreased. We think this is because the HMMs contributed to enhance the state estimation

⁴All these pages were written in Japanese.

⁵Ontologies are provided with their *unique attributes* which are frequently used to represent the entities. An ontology for a human, for example, is provided with the attribute *hobby*. We can select ontologies according to those unique attributes.

| PC specification pages (# of roles = 20) | | | |
|--|------|-------|-----------|
| Method | Rec. | Prec. | F-measure |
| HMM | 0.68 | 0.36 | 0.47 |
| Naive Bayes | 0.54 | 0.41 | 0.47 |
| Company profile pages (# of roles = 20) | | | |
| Method | Rec. | Prec. | F-measure |
| HMM | 0.58 | 0.44 | 0.50 |
| Naive Bayes | 0.42 | 0.44 | 0.43 |
| About-me pages (# of roles = 20) | | | |
| Method | Rec. | Prec. | F-measure |
| HMM | 0.51 | 0.43 | 0.47 |
| Naive Bayes | 0.45 | 0.47 | 0.46 |
| About-me pages (# of roles = 60) | | | |
| Method | Rec. | Prec. | F-measure |
| HMM | 0.41 | 0.46 | 0.43 |
| Naive Bayes | 0.34 | 0.47 | 0.39 |

Table 1: Results in recall and precision; Rec. means recall and Prec. means precision.

for each block. Whether the state estimation for each block is correct or incorrect, its effects/side-effects are extended to neighboring blocks. Precision did not improve because of the effect of the incorrect state estimation extended to neighboring blocks, as well as the correct estimation. We therefore believe that if the accuracy of state estimation by the Naive Bayes Classifier increases, the validity of HMMs will also increase because extension of the correct estimation will be larger than that of the incorrect estimation.

To see the effect of the number of non-rare roles in ontologies, we evaluated the performance of state estimation for about-me pages when 60 non-rare roles were used. The result is also shown in Figure 1. HMMs outperformed the Naive Bayes Classifier by 0.04 in F-measure. Increase of the number of non-rare roles causes the use of relatively non-frequent roles which cannot be properly estimated only from training data. In such cases, estimation relying on the states for neighboring blocks using HMMs becomes more helpful.

5 Conclusion and Future Work

In this paper we proposed a method to extract attributes and values by consulting ontologies extracted from HTML tables. We used the Good Turing Estimation in calculating probabilities, and introduced a rare role and a sentence role to filter out useless blocks in Web pages. We applied HMMs to the problem and proposed the technique to estimate

parameters for the HMMs.

Because the task proposed in this paper is still simple one, it can be extended in several directions. For example, some topic detection techniques might be employed to deal with Web documents on many kinds of topics by using various kinds of ontologies. We also plan to extend our algorithm to deal with multiple entities on a page, or to make use of HTML tag information to improve the extraction accuracy of the system.

6 Acknowledgements

I thank my supervisor, Jun'ichi Tsujii, for his support and valuable advices on my work. I am also grateful to Kentaro Torisawa, Yuka Tateisi, Takashi Ninomiya, Yoshimasa Tsuruoka, Yusuke Miyao, Naoki Yoshinaga and four anonymous reviewers for their helpful comments.

References

- H. H. Chen, S. C. Tsai, and J. H. Tsai. 2000. Mining tables from large scale HTML texts. In *Proc. of 18th COLING*, pages 166–172.
- G. D. Forney. 1973. The viterbi algorithm. *Proc. IEEE*, 61:268–278.
- D. Freitag. 1998. Information extraction from HTML: Application of a general machine learning approach. In *Proc. of AAAI-98*, pages 517–523.
- I. J. Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264.
- C. Hsu and M. Dung. 1998. Generating finite-state transducers for semistructured data extraction from the web. *Journal of Information Systems*, 23-8:521–538.
- S. Kurohashi and M. Nagao. 1998. Japanese morphological analysis system JUMAN version 3.5. *Department of Informatics, Kyoto University, (in Japanese)*.
- N. Kushmerick, D.S. Weld, and R. Doorenbos. 1997. Wrapper induction for information extraction. In *Proc. of IJCAI-97*, pages 729–735.
- I. Muslea, S. Minton, and C. Knoblock. 1999. A hierarchical approach to wrapper induction. In *Proc. of the third International Conference Autonomous Agents*, pages 190–197.
- M. Yoshida, K. Torisawa, and J. Tsujii. 2001. Extracting ontologies from World Wide Web via HTML tables. In *Proc. of PAFLING 2001*, pages 332–341.