Automated data extraction from the web with conditional models

Xuan-Hieu Phan*

Graduate School of Information Science,
Japan Advanced Institute of Science and Technology (JAIST),
1-1, Asahidai, Nomi, Ishikawa 923-1292 Japan
E-mail: hieuxuan@jaist.ac.jp
*Corresponding author

Susumu Horiguchi

Graduate School of Information Sciences,
Tohoku University, Aoba 6-3-09, Sendai 980-8579, Japan
E-mail: susumu@ecei.tohoku.ac.jp

Tu-Bao Ho

Graduate School of Knowledge Science,
Japan Advanced Institute of Science and Technology (JAIST),
1-1, Asahidai, Nomi, Ishikawa 923-1292 Japan
E-mail: bao@jaist.ac.jp

Abstract: Extracting data on the Web is an important information extraction task. Most existing approaches rely on wrappers which require human knowledge and user interaction during extraction. This paper proposes the use of conditional models as an alternative solution to this task. Deriving the strength of conditional models like maximum entropy and maximum entropy Markov models, our method offers three major advantages: the full automation, the ability to incorporate various non-independent, overlapping features of different hypertext representations, and the ability to deal with missing and disordered data fields. The experimental results on a wide range of e-commercial websites with different layouts show that our method can achieve a satisfactory trade-off between automation and accuracy, and also provide a practical application of automated data extraction from the Web.

Keywords: web mining; information extraction; statistical machine learning; maximum entropy; maximum entropy Markov model; conditional model.

Reference to this paper should be made as follows: Phan, X-H., Horiguchi, S. and Ho, T-B. (2005) 'Automated data extraction from the web with conditional models', *Int. J. Business Intelligence and Data Mining*, Vol. 1, No. 2, pp.194–209.

Biographical notes: Xuan-Hieu Phan graduated from the Faculty of Technology, Vietnam National University, Hanoi in Computer Science in 2001. He received his Master's degree from the same university in 2003. He is now a PhD student at the Graduate School of Information Science, JAIST. His research interests have been mainly concerned with Data Mining

(Association Rules, Text and Web Mining), Natural Language Processing, Information Extraction and Statistical Machine Learning.

Susumu Horiguchi received his BS, MS and Doctoral degrees from Department of Communication Engineering, Tohoku University in 1976, 1978, and 1981, respectively. He was a Visiting Scientist at IBM Thomas J. Watson Research Centre from 1986 to 1987 and a Visiting Professor at University of Southwestern Louisiana and Texas A&M University (summer in 1994 and 1997). Currently, he is a Full Professor in Tohoku University. He has been involved in organising many international conferences sponsored by IEEE, IEICE, and IPS. His research interests include interconnection and optical networks, parallel and distributed computing, VLSI/WSI architecture, and data mining and KDD.

Tu-Bao Ho has been a Full Professor at School of Knowledge Science, Japan Advanced Institute of Science and Technology (JAIST), since 1998. He received his BTech in Applied Mathematics from Hanoi University of Technology (1978), MS and Doctoral degrees in Computer Science from Pierre and Marie Curie University, Paris in 1984 and 1987, respectively. He was a Research Fellow (1983–1987) at French National Institute for Research in Computer Science and Control, Visiting Fellow (1992) at Wiscosin-Madison University (USA). His research interests include knowledge-based systems, machine learning, and data mining and KDD.

1 Introduction

Information extraction (IE) can be defined as the process of extracting segments from semistructured or free text to fill data slots/fields in a predefined record template. As a particular subdirection of NLP, IE was originally used to find specific information from natural language documents such as named entities, elements, coreferences, relations and scenario (Grishman and Sundheim, 1995). However, with the huge volume of data residing on the web, IE is also considered as the task of extracting desired information in different hypertext formats to populate relational databases.

Several approaches, such as wrapper based, NLP based and ontology based methods, have been employed to extract data records from the web. Wrapper based tools, like WIEN (Kushmerick, 2000), SoftMealy (Hsu and Dung, 1998), Stalker (Muslea et al., 2001) and DEbyE (Laender et al., 2002), build wrappers based on objects of interest from sample pages to get extraction rules which are, in turn, used to extract similar objects from similar pages. Although these tools usually achieve high accuracy, they have several drawbacks.

- they require user knowledge and user intervention to mark objects of interest in the sample pages, and it is inconvenient for normal users to extract data from a huge set of pages with various formats or from unfamiliar domains
- wrappers are sensitive to the change of web page structures, which often occurs on the web.

NLP based tools such as RAPIER (Califf and Mooney, 1999), SRV (Freitag, 2000) and WHISK (Soderland, 1999) usually use traditional NLP techniques, such as text chunking and 'part of speech' (POS) tagging, to learn rules for extracting desired data from highly grammatical documents; however, these tools are not as suitable for less grammatical web pages. In the ontology based technique (Embley et al., 1999), an ontology is previously constructed to describe the data of interest, including taxonomies, relationships and lexical entries. By parsing this ontology, the tool can automatically produce a database by recognising and extracting data present in pages given as input. However, this approach is still labour intensive in building and maintaining the ontologies.

In this paper, we propose the use of conditional models as a statistical machine learning approach for automatically integrating data on the web. The two conditional models we employ in this paper are MaxEnt (Berger et al., 1996) and MEMM (McCallum et al., 2000). MaxEnt is a statistical model that has been successfully used for various NLP tasks such as POS tagging (Ratnaparkhi, 1996, 1998), named-entity recognition (Borthwick, 1999; Chieu and Ng, 2002), machine translation (Berger et al., 1996) and data integration (Phan et al. 2004b). MEMM, a kind of conditionally trained finite state machine (FSM), combines the idea of MaxEnt and the first order Markovian property to form a sequential tagging model in which the probability of reaching the current state depends on both the current data observation and the previous state.

In our work, data slots/fields in a record template are predefined via a number of tags or labels, and the conditional models are trained to classify sequences of hypertext/data segments to fill these slots. The whole process is as follows. First, the input web page is parsed to build an HTML tree. Then, we locate data regions containing data records by estimating the Shannon's entropy at each internal node. Found records are transformed into sequences of data segments. Next, various features at different levels (vocabulary, capitalisation, HTML tags, semantics) in segments are integrated into the conditional models to utilise the rich contextual information. This is known as the feature selection step. Finally, the trained conditional models classify segments to fill record templates. In this sense, our method can be thought of as a sequential tagging application.

The major contribution of our work is three fold:

- our method can make the most of various kinds of contextual information from hypertext documents; in other words, it can integrate a large number of nonindependent, overlapping features at different levels of granularity
- it can deal with missing values or disorder problems that are the pitfalls in wrapper based methods; this is because the tag of a hypertext segment depends only on its own information and does not conform to any prespecified order
- once trained, our models will automatically extract data without any user interaction; this full automation is a big convenience for nonexpert users who wish to extract data from a huge volume of web pages or from unfamiliar information sources.

The remaining part of the paper is organised as follows. Section 2 presents the background of the two conditional models. Then, the whole framework and the details of the proposed approach are discussed in Section 3. Section 4 presents the experimental results and some discussion. Finally, Section 5 makes conclusions and states the future work.

2 Conditional models

MaxEnt (Berger et al., 1996) is an approach to build a classifier around an estimated distribution. The underlying idea of MaxEnt is to use everything that we know from the data, but assume nothing else. In other words, MaxEnt is the model having the highest entropy while it is compatible with constraints derived from the empirical data. MEMM (McCallum et al., 2000) is built on top of the MaxEnt model by combining both the underlying idea of MaxEnt and the first order Markovian property.

2.1 Maximum entropy

Given: a training data set $D = \{(o_1, s_1), (o_2, s_2), \ldots, (o_Q, s_Q)\}$ where o_i is the data observation and s_i is the corresponding tag (also called label or class). Conditional MaxEnt is a conditional distribution in the form of $P(s / o)$ – the conditional probability of tag s, given the observation o. This model will be used to classify future observations. To learn from the training data, experimenters have to determine significant features from the training data and integrate them into the MaxEnt model in terms of constraints. Features selected from the training data are useful facts and usually have the form of a two argument function $f:(o, s) \rightarrow R$.

$$f_{<cp,s'>}(o,s) = \begin{cases} 1 & \text{if } s = s' \text{ and } cp(o) = \text{true} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where s' is a tag and cp is a context predicate that carries a piece of useful contextual information about observation o. In general, context predicate cp in equation (1) is an arbitrary predicate that represents a useful characteristic of the observation. The expected value of a feature f_i with respect to the empirical distribution $\tilde{P}$, denoted as $E_{\tilde{P}} f_i$, is exactly the number of times feature f_i is observed in the training data $E_{\tilde{P}} f_i = \sum_{(o,s)\in D} \tilde{P}(o,s) f_i(o,s)$. The expected value of the feature f_i with respect to the conditional MaxEnt model $P(s \mid o)$ is defined as $E_P f_i = \sum_{(o,s)} \tilde{P}(o) P(s \mid o) f_i(o,s)$. The MaxEnt model is consistent with the training data with respect to the feature f_i; thus we have the following constraint.

$$E_P f_i = E_{\tilde{P}} f_i. \quad (2)$$

If we want to encode k features into the model, then we will have k constraints like equation (2). The MaxEnt model is the model $P(s \mid o)$ that has the highest entropy while satisfying k above constraints. By applying the method of Lagrange multipliers from the theory of constrained optimisation, (Pietra et al., 1997) proved that MaxEnt has the following exponential form and, furthermore, the found model is unique and agrees with the maximum likelihood distribution.

$$P_\lambda(s \mid o) = \frac{1}{Z_\lambda(o)} \exp \sum_i \lambda_i f_i(o,s) \quad (3)$$

where λ_i is the Lagrange multiplier associated with feature f_i, and

$$Z_\lambda(o) = \sum_s \exp\left(\sum_i \lambda_i f_i(o,s)\right)$$

is the normalising constant to ensure that $P_\lambda(s \mid o)$ is a distribution. The solution to the MaxEnt model is also the solution to a dual maximum likelihood problem. Further, it is guaranteed that the likely surface is convex, having a single global maximum. The MaxEnt model is most commonly trained using Generalised Iterative Scaling (Darroch and Ratcliff, 1972). Other algorithms, such as Improved Iterative Scaling (Pietra et al., 1997), are often used to speed up the training phase.

2.2 Maximum entropy Markov model

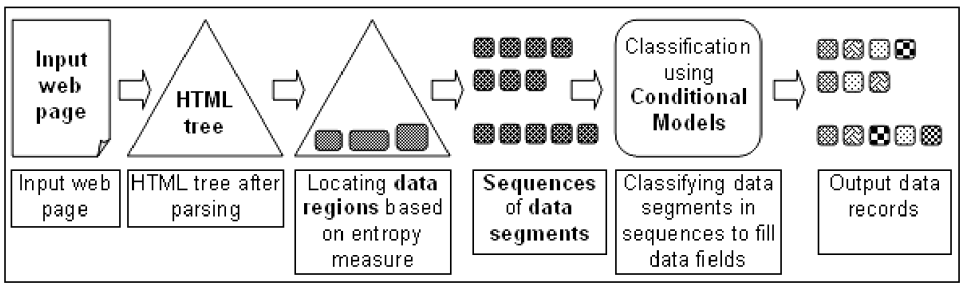
MEMM is similar to HMM except that the transition probability $P(s \mid s')$ and the emission probability $P(o \mid s)$ are replaced by a single probability $P(s \mid s', o)$ – the probability of the current state s, given the previous state s' and the current observation o. $P(s \mid s', o)$ is the MaxEnt model corresponding to s' and also has the exponential form: $P_{s'}(s \mid o) = 1/Z(o,s')\exp\left(\sum_i \lambda_i f_i(o,s)\right)$. MEMM is actually a chain of $|S|$ MaxEnt models where S is the set of all states. To train MEMM, we split the original training data into $|S|$ parts and then apply an iterative scaling algorithm (e.g., GIS, IIS) to train each MaxEnt separately. The decoding of MEMM is similar to that of HMM, using the Viterbi algorithm with *forward* $\alpha_t(s)$ or *backward variables* $\beta_t(s)$. Space limitations prevent a detailed discussion of MEMM; refer to McCallum et al. (2000) for a full description.

3 The proposed approach

This section presents our approach using the conditional models mentioned above to extract data records from hypertext documents. Figure 1 depicts the overall framework for extracting data records from the web that includes two main phases:

- locating data regions and sequences of hypertext/data segments from input web pages
- classifying data segments by using conditional models to fill data fields in output data records.

Figure 1 Overall framework for extracting data records from the web



The first phase parses each input web page to form the corresponding HTML tree including HTML tags, formats, images, and free text. Then, data regions (if existing) will be located using an entropy estimation to measure the similarity among HTML subtrees. Found data regions are then divided into sequences of data segments that are, in turn, the inputs of the second phase. The second phase classifies data segments to fill data fields of a predefined record template. In order to achieve an accurate classification, this phase employs two conditional models (MaxEnt and MEMM) that were trained with various types of contextual information observed in the training data.

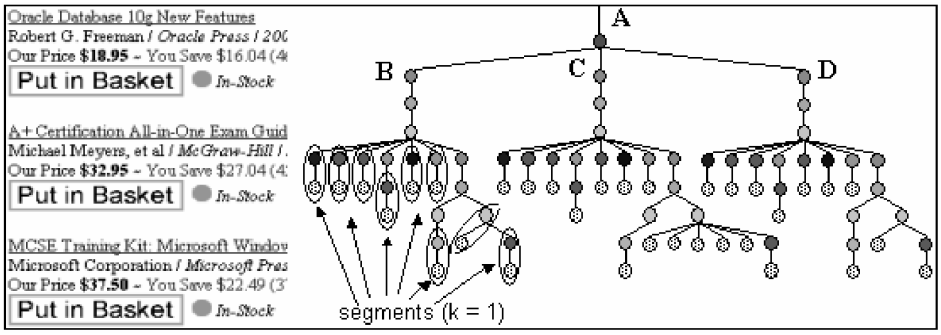
The following subsections discuss the proposed framework in detail:

- locating data regions from input pages
- the building of conditional models based on four types features
- the models training, decoding, and testing.

3.1 Locating data regions from input web pages

This section mainly describes the use of Shannon's entropy estimation to identify data records from an input web page. This idea originates from the observation that a group of data records containing descriptions of a set of similar objects are typically presented in a contiguous data region of a page and are formatted using similar HTML tags. Figure 2 shows an example of a data region containing three records and its HTML hierarchical structure. We see that data records reside in three *similar* subtrees. The term *similar*, in this sense, means these subtrees have structures that are analogous in both tree skeleton and tag position. If we can map HTML subtrees into a set of *representative values* (RVs) that reflects their structures, then the similarity among these subtrees can be measured by calculating Shannon's entropy on the set of RVs. A data region tends to have a high entropy value because its subtrees have similar RVs. For example, node A in Figure 2 should have a high entropy because its subtrees B, C and D are very similar in structure. Thus, A is recognised as a data region.

Figure 2 A data region containing three records and its HTML hierarchical structure



We can find any mapping to map a tree structure into a RV provided that the mapping satisfies the condition that two similar subtrees will have similar RVs, otherwise they will have different values. We propose a simple but efficient mapping as follows. The RV of a subtree T (T is also the root node), denoted as $T.rv$, is calculated by the formula:

$$T.rv = T.tw + \sum_{N_i \in N} (N_i.tl \times N_i.co \times N_i.tw),$$

where N is the set of all descendant nodes of T. $N_i.tl$ is the tag level, i.e., the distance from tag node N_i to the root node T. $N_i.tw$ is the tag weight of the tree node N_i. The main usage of $N_i.tw$ value is to help distinguish among different HTML tags. $N_i.co$ is the child order of the node N_i among its siblings.

After Shannon's entropy is estimated at all internal nodes of the HTML tree, data regions will be located if their normalised entropy values ($\in [0,1]$) exceed a given minimum threshold. Then, a set of heuristic rules is used to filter noisy regions that do not contain real data records. The details of the algorithms and explanations are presented in Phan et al. (2004a). Each found data record is a HTML subtree, and the real contents (images and text) are located at leaf nodes. The HTML subtree is traversed according to preorder. At each leaf node, the contents of the leaf together with the contents of its k ancestral nodes are copied to constitute a hypertext/data segment, see Figure 2. In this way, each data record is transformed into a sequence of data segments. Figure 3 presents the sequence of eight segments (with $k = 1$) of the first record in Figure 2. In this sequence, the text in the pair of square brackets is the tag name. Tag information is added manually when we prepare the training data. The second column is the content of the leaf node, and the third column is the content of the parent node. Sequences of data/hypertext segments then act as inputs for the second phase for classification.

Figure 3 The sequence of eight hypertext segments of the first record in Figure 2

Tag	Leaf node	Parent node
[PN]	<text>Oracle Database 10g New Features	<a href="...0072229470">
[PA]	<text>Robert G. Freeman /	<td valign="top">
[PM]	<text>*Oracle Press / 2004 / 0072229470*	<i>
[PP]	<text>Our Price	<td valign="top">
[PP]	<text>**$18.95** ~	<b>
[PYS]	<text>You Save $16.04 (46% Off)	<font color="#990000">
[NNI]	<img src="...in.gif" alt="..." height="13"...>	<td>
[PS]	<text>*In-Stock*	<i>

PEWeb, our tool for locating data regions and sequences of data segments, is available at www.jaist.ac.jp/~hieuxuan/softwares/peweb/. This tool makes online queries to retrieve web pages, then parses them to create HTML trees, and finally locate data regions and sequences of data segments based on an entropy measure.

3.2 Feature selection for the conditional models

In our approach, four kinds of features are integrated into the models: HTML tag, capitalisation, vocabulary, and semantics. HTML related features take advantage of HTML tag information such as <font>, <a>, etc. Capitalisation related features focus on uppercase and lowercase of text. Vocabulary related features make the most of various salient words, collocations, or phrases such as 'IBM', 'Emily Bronte', 'our price', etc. The semantic features are based on human observations. Almost all features are generated automatically from the training data via the set of feature templates listed in

Table 1. A small number of special features such as semantic features are added to the model based on human perception.

Table 1 Several examples of feature template

Type	*Context predicate*	*Tag*
HTML	o_i has HTMl tag x	$s_i = T$
	o_i has image with height ∈ [a, b] and width ∈ [c, d]	$s_i = T$
	o_i has image type *.jpg, *.gif, etc.	$s_i = T$
Vcb	o_i has word or phrase w	$s_i = T$
	o_{i-1} has word or phrase w	$s_i = T$
	o_i has $, %, etc.	$s_i = T$
	Number of digits that o_i has ∈ [a, b]	$s_i = T$
	TextLength (o_i) ∈ [a, b]	$s_i = T$
Cap	CapAll (o_i) = TRUE and textlength(o_i) ∈ [a, b]	$s_i = T$
	InitCap (o_i) = TRUE and textlength(o_i) ∈ [a, b]	$s_i = T$
	LowerAll (o_i) = TRUE and textlength(o_i) ∈ [a, b]	$s_i = T$

HTML tag related features (Html). These features are related to HTML tags and their characteristics. Any contextual information about HTML tags can be integrated providing it is highly discriminative. For example, the name of a product usually has a hyperlink tag; the selling price is usually formatted using a bold tag (<b>); the discount information is usually highlighted with bright colour using a font tag (<font...>); The *alt* or *title* attribute of the representative image (PI) is usually similar to the product name (PN), etc. Information about HTML tags is a rich and useful source of evidence for the models. The following example feature emphasises the fact that an observation o is the product name field while it has a hyperlink tag:

$$f_{<hyperlink,PN>}(o,s) = \begin{cases} 1 & \text{if } s = PN \text{ and } o \text{ has <a href …>} \\ 0 & \text{otherwise} \end{cases}.$$

Table 2 Examples of HTML-related features

Context predicate	*Tag*
o has hyperlink tag <a href …>	PN
o has font tag <font colour = …>	PYS
o has image with height ∈ [100, 200]	PI
o has bold tag <b>	PP
o has image with height ∈ [1, 20]	NNI
o has image type *.jpg	PI

Capitalisation related features (Cap). Capitalisation is another useful source of evidence. According to our observation, product name, author name and manufacturer are usually capitalised at the first character of each word. Capitalisation related information, by itself, is not highly discriminative. However, it is useful if it is combined with other kinds of information such as length of text.

Vocabulary related features (Vcb). Vocabulary is one of the most important sources of contextual information. Words carry much discriminative information. For example, the appearance of the words or phrases 'price', 'your price', 'list price', or 'asking' in an observation is strong evidence of price related fields; the presence of 'save', 'you save', 'off' indicates the discount; the appearance of 'found', 'empty', or 'in stock' indicates the status field; the presence of 'by', 'written by', 'author', 'produced by', 'Emily Bronte', 'Sony', or 'Macintosh' indicates the author of a book or the manufacturer of a product. The following example highlights the fact that if an observation contains the phrase 'Our Price', it should be related to the sale price (PP):

$$f_{<\text{'Our Price', PP}>}(o,s) = \begin{cases} 1 & \text{if } s = PP \text{ and } o \text{ contains 'Our Price'} \\ 0 & \text{otherwise.} \end{cases}$$

In addition to vocabulary, digits, numbers, and special symbols are also valuable sources of contextual information. For example, the appearance of dollar sign ($), *USD*, *EURO*, or *Yen* in an observation indicates the price field; the presence of percent sign (%) indicates the discount percentage; the existence of an integer number together with a % sign is strong evidence of a discount percentage. Table 3 shows several examples of vocabulary related features. The following is an example of number related features.

$$f_{<\text{a \% sign and an integer, PYS}>}(o,s) = \begin{cases} 1 & \text{if } s = \text{PYS and } o \text{ has \% sign and an integer number} \\ 0 & \text{otherwise.} \end{cases}.$$

Table 3 Examples of vocabulary related features

Context predicate	*Tag*
o has word 'our price'	*s* = PP
o has word 'outlet price'	*s* = POP
o has word 'empty'	*s* = PS
o has % sign	*s* = PYS
o has word 'Conan Doyle'	*s* = PA
o has % sign and an integer	*s* = PYS
o has word 'list price'	*s* = PLP
o has word 'you save'	*s* = PYS
o has $ sign	*s* = PP
o has word 'Oracle'	*s* = PM
o has $ sign and a real number	*s* = PP

Semantic features (Smt). This kind of feature provides additional information to solve ambiguous situations. For example, a record contains two price fields ($80.50 and $87.60). How can we determine which is the selling price and which is the list price? In the real world, the selling price is usually smaller than the list price, and the outlet price is usually less than the retail price. The following feature says that if the current observation o_i contains a real number r_i and this number is greater than a real number r_i in another observation o_j (o_i and o_j in the same sequence), then o_i should be the list price.

$$f_{<r_i, r_j, \text{PLP}>}(o_i, s_i) = \begin{cases} 1 & \text{if } s_i = \text{PLP and } r_i \in o_i \text{ and } r_j \in o_j \text{ and } r_i < r_j \\ 0 & \text{otherwise.} \end{cases}$$

3.3 Model learning, decoding, and testing

Learning: Training data is a set of sequences of segments. Each sequence corresponds to a data record, and each segment belongs to, at most, one field. All sequences in the training data set are labelled as shown in Figure 3. Then, all of the features described in Section 3.2 are generated automatically from the training data via the feature templates listed in Table 1. Features with a frequency in the training data larger than a given threshold (5 in our experiments) are retained. Other features are discarded because they are not confident enough. Finally, the MaxEnt and MEMM models are trained, using the GIS algorithm.

Decoding. Given a sequence of annotated segments, the models will classify all the segments in the sequence. For the MaxEnt model, we compute the set of conditional probability $P_\lambda(s|o)$ in formula (equation (3)) for each tags in Table 4. The tag corresponding to the highest probability will be the tag for segment o. This process is repeated for all segments in the sequence. For the MEMM model, we use the Viterbi algorithm to search the most likely tag sequence for the sequence of segments.

Table 4 Tags of data fields/slots in the record template

Tag	*Description*
PI	Representative image of product
PN	Name of product
PLP	List price of product
PRP	Retail price of product
PA	Author of product (e.g., book)
PS	Status of product (e.g., empty)
NNT	Other or noisy text
NNI	Other or noisy image
PP	Selling price of product
POP	Outlet price of product
PYS	Discount money
PM	Manufacturer of product
PD	Description of product

Testing. After decoding, we compare the 'model generated' tags and the manually annotated tags in the sequence to count the number of errors. The numbers of correctly and incorrectly classified sequences (records) as well as the number of true data records available on the input page are counted to calculate precision and recall.

4 Evaluation

4.1 Experimental setting

In this section, we evaluate the experimental results of our system that includes two modules: PEWeb-ME (for MaxEnt) and PEWeb-MEMM (for MEMM). We also compare our system with a wrapper based tool – DEByE (Laender et al., 2002). This is the state of the art system for extracting data from the web; DEByE was also compared

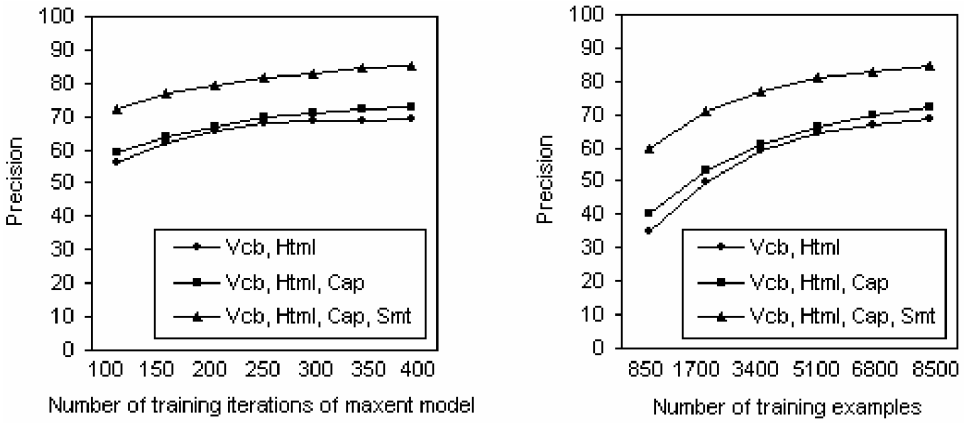
with SoftMealy, WIEN, and Stalker in Laender et al. (2002). Because DEByE is a wrapper based tool, each time it extracts data from a new website, it requires example objects in sample pages from the user in order to build wrappers or extraction rules. These rules, in turn, are used to extract similar objects from similar pages. This means that DEByE requires the user knowledge and user interaction during extraction. For web pages with uniform and regularly formatted data records, DEByE needs only one or two example records. For loosely formatted pages or pages with various kinds of data records, DEByE requires more examples to achieve a high recall. Our method, on the other hand, requires a set of manually annotated data to train the system. However, once trained, our system automatically extracts data from new web pages without any user interaction. This is the biggest difference between the two systems.

The training and testing data for our system are also available at www.jaist.ac.jp/~hieuxuan/softwares/pewebme/. They include approximately 2000 sequences (10,000 segments). These data sets are extracted from various e-commercial sites such as amazon.com, ebay.com, and compusa.com. 15% of this set was chosen randomly for testing and the rest is the training set.

4.2 Experimental results and discussion

Figure 4 shows the precision of the MaxEnt as a function of kinds of features and the number of training iterations as well as the size of training data. In this experiment, the number of iterations was gradually increased to tune the optimal set of parameters. This also demonstrates that performance strongly depends on kinds of features and that semantic related features play an important role in enhancing the precision of the system. This figure also depicts the dependence of precision on the size of the training data. In this experiment, 15% of the annotated data set was chosen randomly for testing and the remaining part was used for training. The figure shows that the precision changes significantly when the size of the training data is increased. This is reasonable because conditional models usually need a relatively large training set to work efficiently.

Figure 4 Precision depends on number of training iterations and size of training data



The experimental results comparing our system and DEByE for a number of commercial websites are described in Table 6. The first column is the ID numbers of the websites listed in Table 5. The second column is the actual number of data records available in the web page. The third and fourth columns are the number of found and correctly found records for the PEWeb-ME module, and the next two columns are the precision (#C./#F.) and the recall (#C./#Rec.) percentage for PEWeb-ME. The next four columns are similar experimental results for PEWeb-MEMM module. The same set of e-commercial websites was also extracted using the DEByE system. The eleventh column (#Ex.) is the number of example records marked by the user to build wrappers. The twelfth column (#C.) is the number of correct records retrieved by DEByE, and the last column is the recall (#C/#Rec.) of DEByE. We did not measure the precision of DEByE like (Laender et al. 2002). This is reasonable because DEByE is a user driven tool and what it extracts is usually consistent with desired information. The last two lines of Table 6 are the average performances (precision and recall) of PEWeb-ME, PEWeb-MEMM, and DEByE calculated from 20 websites mentioned in Table 5. The first average (i.e., Avg1.) is calculated from the sum of 20/site precision and recall whereas the second average (i.e., Avg2.) is the absolute numbers of #Rec., #F., and #C. of 20 websites. Generally speaking, Avg1. reflects the extent to which a system can adapt to different web structures and formats while Avg2. reflects the total performance of a system with respect to the total number of data records that are correctly extracted. The highest Avg1. precision and recall of our system are 86.56% and 88.60% (PEWeb-ME) comparing to the highest Avg1. DEByE recall of 96.10%. The highest Avg2. precision and recall of our system are 89.92% and 92.69% (PEWeb-ME) comparing to the highest Avg2. DEByE recall of 95.45%.

Table 5 Experimental websites

No.	*Website*
1	shopping.yahoo.com (MP3 player)
2	www.overstock.com (Furniture)
3	art.listings.ebay.com (digital art)
4	www.amazon.com (DVD Players)
5	www.kidsfootlocker.com (shoes)
6	software.thepricesearch.com (soft)
7	chemstore.cambridgesoft.com (soft)
8	www.compusa.com (Apple Mac)
9	qualityinks.com (soft)
10	www.ubid.com (monitor)
11	www.nothingbutsoftware.com (soft)
12	www.radioshack.com (hardware)
13	www.softwareoutlet.com (soft)
14	www.sephora.com (cosmetic)
15	www.etoys.com ($40–$50)
16	www.drugstore.com (beauty and spa)
17	www.onsale.com (general)
18	www.target.com (general)
19	shop.lycos.com (sunglasses)
20	www.nexttag.com (flower \and plant)

Table 6 Precision and recall comparison

Web site	*# Rec.*	*PEWeb-ME* (*MaxEnt*)				*PEWeb-MEMM*				*DEByE*		
		#F.	*#C.*	*Pr.* (%)	*Rc.* (%)	*#F.*	*#C.*	*Pr.* (%)	*Rc.* (%)	*#Ex.*	*#C.*	*Rc.* (%)
1	15	15	13	86.7	86.7	15	13	86.7	86.7	1	15	100.0
2	24	24	21	87.5	87.5	24	18	75.0	75.0	2	22	91.6
3	50	55	49	89.1	98.0	55	50	91.0	100.0	3	46	92.0
4	25	26	21	80.7	84.0	26	20	76.9	80.0	2	22	88.0
5	12	12	12	100.0	100.0	12	12	100.0	100.0	2	12	100.0
6	10	10	10	100.0	100.0	10	10	100.0	100.0	1	10	100.0
7	222	219	215	98.2	96.8	219	205	93.6	92.3	4	214	96.4
8	16	16	14	87.5	87.5	16	16	100.0	100.0	1	16	100.0
9	37	39	34	87.2	91.9	39	32	82.1	86.5	4	35	94.6
10	75	85	72	84.7	96.0	85	70	82.4	93.3	6	71	94.7
11	13	14	10	71.4	76.9	14	10	71.4	76.9	1	12	92.3
12	3	3	3	100.0	100.0	3	3	100.0	100.0	1	3	100.0
13	11	10	8	80.2	72.7	10	8	80.0	72.7	1	11	100.0
14	9	9	7	77.8	77.8	9	6	66.7	66.7	1	9	100.0
15	20	23	16	69.6	80.0	23	13	56.5	65.0	2	19	95.0
16	14	16	13	81.3	92.9	16	12	75.0	85.7	1	12	85.7
17	21	21	21	100.0	100.0	21	20	95.2	92.2	3	21	100.0
18	12	12	10	83.3	83.3	12	12	100.0	100.	2	12	100.0
19	12	11	8	72.7	66.7	11	7	63.6	58.3	1	11	91.7
20	15	15	14	93.3	93.3	15	15	100.0	100.0	1	15	100.0
Avg1.				86.56	88.60			84.81	86.57			96.10
Avg2.	616	635	571	89.92	92.69	635	552	86.93	89.61		588	95.45

The highest Avg1. precision of 86.56% and recall of 88.60% demonstrate that our system can deal with different websites with various unseen representation styles (i.e., data record layout and order of data fields) and HTML formats (hyperlinks, fonts, text format and colour, etc.) based on the knowledge learned from the training data. In addition, the highest Avg2. precision of 89.92% and recall of 92.69% show that our system can attain a high performance when the number of actual data records (i.e., the size of input web collection) becomes large. For instance, when working with the seventh web page that has a large number of data records (222 actual records), recall of PEWeb-ME (96.8%) is a little bit higher than that of DEByE (96.4%). That is because PEWeb-ME is flexible enough to deal with some slight changes (e.g., the change of record layout, order of fields, etc.) in record structure and layout. However, DEByE fails to recognise such changes because its extraction rules are fixed. The experimental results show that our approach can achieve a high performance without user interaction during extraction.

The average recall of DEByE is about eight (Avg1.) and three (Avg2.) percentage larger than those of our system. There are several reasons for this difference.

- For DEByE, users specify what should be extracted via example data records. The user tells DEByE what it has to do, while our system automatically determines what it should extract. And thus, the knowledge it learns from the training data obviously cannot deal with all situations in new environments.
- The vocabulary related features integrated in our models are all derived from the training data via feature templates. 10,000 training examples, of course, cannot cover all of the highly discriminative words used on the web; if a segment contains no familiar words and it does not gain sufficient support from other kinds of features (i.e., Html, Cap, Smt), our system may fail to classify it.
- Feature selection is the most important step influencing the precision and recall factors. A wise strategy for feature selection means a significant increase in system performance.

The experimental results also show that the performance of PEWeb-MEMM is lower than that of PEWeb-ME. This is because the sequential dependence among data slots is not as strong as that of other sequential tagging tasks like part of speech. Furthermore, MEMM usually needs more training data than MaxEnt to achieve a high performance.

5 Conclusions and future work

This paper presents a new approach using conditional models for extracting commercial data from the web. We began by introducing the theoretical aspect of conditional models (i.e., MaxEnt and MEMM) and their strengths in dealing with sequential tagging and information extraction problems. We next presented the proposed approach for the fully automated extraction process that includes two main phases:

- locating data regions and data segments from a web page
- classifying those data segments to fill data fields in a record template.

In the first phase, data regions were located in an input web page based on an entropy estimation among representative values of HTML subtrees. The second phase took data segments extracted from those HTML subtrees as inputs and classified them to fill data fields. The classification using conditional models, integrated many useful features ranging from HTML format, capitalisation, vocabulary to highly semantic ones. This allows our model to take advantage of various types of contextual information residing in data segments to successfully recognise highly ambiguous data fields on the web environment. Also, our method is flexible in dealing with missing and disordered data fields that are pitfalls in wrapper based methods. Last but not least, once trained, our model can automatically extract data from input raw web pages without any user intervention. This is a big convenience for nonprofessional users who wish to extract data from unfamiliar and very large information sources.

Future work will try to enhance the precision and recall for our system. Feature selection will be refined by using both human perceptions and feature selection algorithms. The use of dictionaries for unseen words or phrases may be an efficient way to reduce classification errors. Finally, more natural language processing techniques such as text chunking and named entity recognition will be employed to preprocess data and make the system more intelligent.

Acknowledgement

This work was in part supported by the Grant-in-Aid of Scientific Research, JSPS, Japan. We would also like to thank the reviewers of this paper who offered invaluable comments and suggestions.

References

Berger, A., Pietra, S.A.D. and Pietra, V.J.D. (1996) 'A maximum entropy approach to natural language processing', *Computational Linguistics*, Vol. 22, No. 1, pp.39–71.

Borthwick, A. (1999) *A Maximum Entropy Approach to Named Entity Recognition*, PhD dissertation, Department of Computer Science, New York University, USA.

Califf, M. and Mooney, R.J. (1999) 'Relational learning for pattern-match rules for information extraction', *Proc. of the 16th National Conf. on AI*, AAA Press/The MIT Press, Florida, USA, 18–22 July, pp.328–333.

Chieu, H.L. and Ng, H.T. (2002) 'A maximum entropy approach to information extraction from semi-structured and free text', *Proc. of the 18th National Conf. on AI*, AAAI Press, Alberta, Canada, 28th July – 1st August, pp.786–791.

Darroch, J.N. and Ratcliff, D. (1972) 'Generalized iterative scaling for log-linear models', *The Annals of Mathematical Statistics*, Vol. 43, pp.1470–1480.

Embley, D.W., Campbell, D.M., Jiang, Y.S., Liddle, S.W., Lonsdale, D.W., Ng, Y.K. and Smith, R.D. (1999) 'Conceptual-model-based data extraction from multiple record web pages', *Data and Knowledge Engineering*, Vol. 31, No. 3, pp.227–251.

Freitag, D. (2000) 'Machine learning for information extraction in informal domains', *Machine Learning*, Vol. 39, Nos. 2–3, pp.169–202.

Grishman, R. and Sundheim, B. (1995) 'Message understanding conference-6: a brief history', *Proc. of the MUC-6*, Morgan Kaufmann Publishers, November, Maryland, USA.

Hsu, C.N. and Dung, M.T. (1998) 'Generating finite-state transducers for semi-structured data extraction from the web', *Information Systems*, Vol. 23, No. 8, pp.521–538.

Kushmerick, N. (2000) 'Wrapper induction: efficiency and expressiveness', *Artificial Intelligence Journal*, Vol. 118, Nos. 1–2, pp.15–68.

Laender, A.H.F., Ribeiro-Neto, B. and Da Silva, A.S. (2002) 'DEByE – data extraction by example', *Data and Knowledge Engineering*, Vol. 40, pp.121–154.

McCallum, A., Freitag, D. and Pereira, F. (2000) 'Maximum entropy markov models for information extraction and segmentation', *Proc. of the 17th ICML*, Morgan Kaufmann Publishers, Stanford, CA, USA, 29th June–2nd July, pp.591–598.

Muslea, I., Minton, S. and Knoblock, C.A. (2001) 'Hierarchical wrapper induction for semi-structured information sources', *Autonomous Agents and Multi-Agent*, Vol. 4, Nos. 1–2, pp.93–114.

Phan, X.H., Horiguchi, S. and Ho, T.B. (2004a) 'PEWeb: product extraction from the web based on entropy estimation', *The IEEE/WIC/ACM Conf. on Web Intelligence*, IEEE Computer Society 20–24 September, Beijing, China, pp.590–593.

Phan, X.H., Horiguchi, S. and Ho, T.B. (2004b) 'Automatic integrating commercial data on the web using conditional models', *ECML/PKDD 2004 Workshop on Statistical Approaches to Web Mining (SAWM04)*, 20–24 September, Pisa, Italy, pp.14–25.

Pietra, S.D., Pietra, V.D. and Lafferty, J. (1997) 'Inducing features of random fields', *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 4, pp.380–393.

Ratnaparkhi, A. (1996) 'A maximum entropy model for part-of-speech tagging', *Proc. of the Empirical Methods in Natural Language Conference*, 17–18th May, Pennsylvania, USA, pp.133–142.

Ratnaparkhi, A. (1998) *Maximum Entropy Models for Natural Language Ambiguity Resolution*, PhD dissertation, School of Information Science, University of Pennsylvania, USA.

Soderland, S. (1999) 'Learning information extraction rules for semi-structured and free text', *Machine Learning*, Vol. 34, Nos. 1–3, pp.233–272.