Extracting Objects from the Web [*]

Zaiqing Nie
Web Search & Mining Group
Microsoft Research Asia
Beijing, P. R. China
t-znie@microsoft.com

Fei Wu[†]
Automation Department
Tsinghua University
Beijing, P. R. China
wufei98@mails.
tsinghua.edu.cn

Ji-Rong Wen Wei-Ying Ma
Web Search & Mining Group
Microsoft Research Asia
Beijing, P. R. China
{jrwen,wyma}@microsoft.com

ABSTRACT
There are various kinds of objects embedded in static Web pages and online Web databases. Extracting and integrating these objects from the Web is of great significance for Web data management. The existing Web information extraction (IE) techniques cannot provide satisfactory solution to the Web object extraction task since objects of the same type are distributed in diverse Web sources, whose structures are highly heterogeneous. The classic information extraction (IE) methods, which are designed for processing plain text documents, also fail to meet our requirements. In this paper, we propose a novel approach called *Object-Level Information Extraction (OLIE)* to extract Web objects. This approach extends a classic IE algorithm, *Conditional Random Fields (CRF)*, by adding Web-specific information. It is essentially a combination of Web IE and classic IE. Specifically, visual information on the Web pages is used to select appropriate atomic elements for extraction and also to distinguish attributes, and structured information from external Web databases is applied to assist the extraction process. The experimental results show OLIE can significantly improve the Web object extraction accuracy.

Keywords
Information Extraction, Wrapper, Conditional Random Field, Information Integration

1. INTRODUCTION
While the Web is traditionally used for hypertext publishing and accessing, there are actually various kinds of objects embedded in static Web pages and online Web databases. There is a great opportunity for us to extract and integrate all the related Web information about the same object together as an information unit. We call these information units *Web objects*. Typical Web objects are products, people, papers, organizations, etc. Commonly, objects of the same type obey the same structure or schema. We can imagine that once these objects are extracted and integrated from the Web, some large databases can be constructed to perform further knowledge discovery and data management tasks.

[*]Technical Report. Microsoft Research. MSR-TR-2004-128
[†]This work is done when the author is visiting Microsoft Research Asia.

This paper studies how to automatically extract object information from the Web. The main challenge is that objects of the same type are distributed in diverse Web sources, whose structures are highly heterogeneous. For instance, information about "paper" objects can be found in homepages, PDF files, and even online databases.

There is no ready method to solve this problem. Instead, there are a series of existing Web information extraction (IE) techniques can provide partial solutions. Specifically, data record detection techniques [18, 34] are proposed to identify data records within a Web page through mining the repeated patterns or templates in HTML codes; attribute value extraction techniques [15, 4, 2, 17, 32] are proposed to further extract the attribute values, also based on template discovery in HTML codes; and attribute value labeling techniques [3, 32] are introduced to label the extracted values with attribute names of the object. Moreover, object identification techniques [30, 6] are then used to integrate all the labeled attribute values from various Web sources about the same object into a single information unit.

Although it is possible to combine these techniques to construct a toolkit to extract object from some template-generated Web pages, we think this is not a practical solution. First of all, since attribute values of an object are extracted from various Web sources independently, it is required to learn a template for each Website. For Web pages which are highly irregular [1] or with few training examples, however, it is impractical to learn such a template for them. Secondly, it is highly ineffective to treat Web information extraction and object identification as two separate phases. Since the accuracy of both extraction and identification suffers, without bi-directional communication between them and the mutually-reinforcing evidence. For example, if an attribute value is wrongly extracted, there is no chance to correct the error in the object identification phase. However, if Web information extraction and object identification are within an

[1]Most Web information extraction techniques rely on the regular patterns in Web pages.

integrated framework, object identification process can interact with the extraction process to improve the extraction accuracy.

Another tightly related work is classic information extraction from plain text document [5, 12, 27, 26, 35]. Integrated inference for extraction and identification has been proposed in some recent works [22, 35]. However, these methods are originally designed for processing plain texts and not for Web pages, and thus cannot be directly applied to the Web object extraction task. Of course, we can transform each Web page into a plain text document by removing HTML tags and other irrelevant codes. But treating Web pages as plain text documents is unwise since some important Web-specific information for object extraction, such as page structure and layout, is lost.

The advantage of classic IE algorithms is their capability of handling heterogeneous data sources and integrating information extraction and object identification in a uniform framework, while Web IE takes advantage of the Web-specific information, e.g. tags and layouts, to extract objects. In this paper, we present an *object-level information extraction (OLIE)* approach which can effectively extract Web objects from multiple heterogeneous Web data sources. Our basic idea is to extend a classic IE algorithm, *Conditional Random Fields (CRF)*, by adding Web-specific features. So our method is essentially a combination of Web IE and classic IE. More specifically, besides text, we found that there are other two kinds of Web information, namely visual information on the Web pages and structured information from Web databases, are of particular importance for Web object extraction.

First of all, for classic information extraction from plain text documents, a fundamental difficulty is to identify the set of words which form a meaningful attribute such as title, name, etc. However, for Web pages, there is much visual information which could be very useful in segmenting the Web pages into a set of appropriate atomic elements instead of a set of words. Typically a Website designer would organize the content of a Web page to make it easy for reading, and semantically related content is usually grouped together. The entire page is divided into regions for different contents using explicit or implicit visual separators such as lines, blank area, image, font size, or colors [36]. For example, the title and authors of a paper in a Web page are usually formatted with different font sizes and put at different positions in the page, and it is easy to visually differentiate them.

Secondly, there is a rich amount of structured information available from the Web databases (or the deep Web [33]). For example, DBLP and ACM Digital Library are two large Web databases containing well-structured data for papers, authors and conferences. These database information could also be very useful in Web object extraction. On the Web, information about objects is usually redundant. For example, you may find the information about a paper or an author from multiple Websites. Therefore, the object extraction system should be able to communicate with external databases to check whether there are some matches between the object elements and the attributes in the databases, and use the matching results to guide the extraction and labelling process. If we find a good match between some elements and the key attributes of a database record, we can say with high confidence that the page contains information related to the record in the database. Then we can use other attributes of the record to help us extract the rest information of the object or rectify wrong labels. Moreover, when new objects are extracted, the information of these objects could be also used to help extracting other objects. Therefore, even when we cannot obtain any external Web databases, we can still adopt this method. Note that this is substantially different from some classic information extraction approaches using dictionaries, which only use the matching degree of a single word and a name entity in the dictionary during the extraction process (see Section 6 for details).

To test the effectiveness of our method, we conducted experiments to extract paper and author objects for a paper search engine. Our experimental results show that, by adding visual features and database features into the CRF algorithm, the extraction accuracy is significantly improved. The main contributions of the paper are:

1. An Web object extraction and integration framework, which considers all the information about an object as an information unit and conducts Web information extraction and object identification at the same time;

2. Using visual information on a Web page to select appropriate atomic elements for extraction and also using visual information to distinguish attributes;

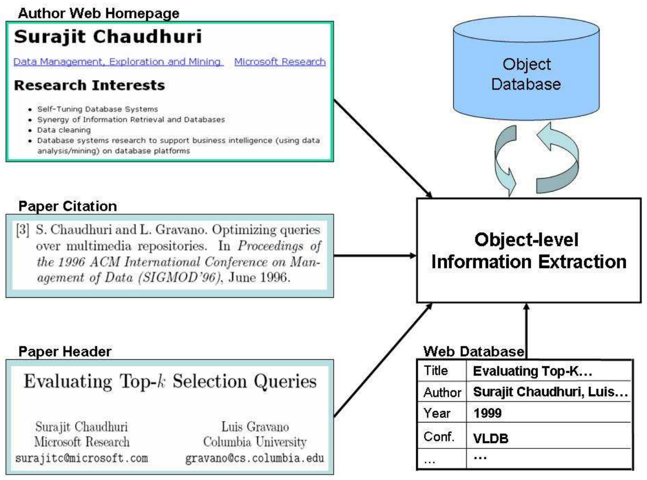
3. Using structured information from external databases to assist the extraction process;

4. An Enhanced CRF model to take into account all the text, visual, and database information during the extraction.

The rest of the paper is organized as follows. In the next section, we formally define the Web object extraction problem and provide some necessary background. Section 3 discusses all types of features that could be used in Web information extraction. Section 4 introduces our Object-Level Information Extraction approach and describes how the basic Conditional Random Fields(CRF) model is extended. Section 5 describes the setting for the experiments we have done to evaluate the effectiveness of our approach, and presents the experimental results. Related work is discussed in Section 6, followed by our conclusions in Section 7.

2. PROBLEM FORMULATION & BACKGROUND

In this section, we first further motivate the Web object extraction problem in the context of a scientific literature search engine that we are developing. We then formally define the problem. Finally we briefly introduce the Conditional Random Fields model which are extended to solve the problem.



Surajit Chaudhuri

Data Management, Exploration and Mining Microsoft Research

Research Interests

- Self-Tuning Database Systems
- Synergy of Information Retrieval and Databases
- Data cleaning
- Database systems research to support business intelligence (using data analysis/mining) on database platforms

[3] S. Chaudhuri and L. Gravano. Optimizing queries over multimedia repositories. In *Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD'96)*, June 1996.

Evaluating Top-k Selection Queries

Surajit Chaudhuri
Microsoft Research
surajitc@microsoft.com

Luis Gravano
Columbia University
gravano@cs.columbia.edu

Title	Evaluating Top-K…
Author	Surajit Chaudhuri, Luis…
Year	1999
Conf.	VLDB
…	…

Figure 1: Web Object Extraction in the Object-level Paper Search Engine

2.1 Motivating Example

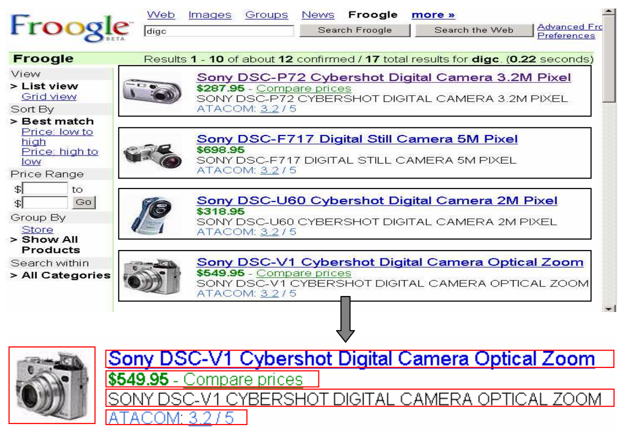
We have been developing an object-level paper search engine called *Libra*[1] [2] (see Figure 1) to help scientists and students locate publication related objects such as papers, authors, conferences, and journals. *Libra* currently indexes 1 million computer science papers crawled from the Web.

Libra collects Web information for all types of objects in the research literature including papers, authors, conferences, and journals. All the related Web information about the same object is extracted and integrated together as an information unit. The objects are retrieved and ranked according to their relevance to the query and their popularity. The object information is stored in an object warehouse with respect to each individual attribute. For example, paper information is stored w.r.t. the following attributes: title, author, year, conference, abstract, and full text. In this way, we can handle structured queries very well, and give different attribute-weights to the hits [8] in different attributes when calculating the relevance score. The details of object relevance and popularity score calculation are beyond the scope of this paper.

Libra extracts information from both the surface Web such as author, conference, and journal homepages and PDF (or PS) files, and the deep Web databases such as DBLP and ACM DL. The data from Web databases can be easily and accurately extracted since most of them are structured and stored in XML files. The data from surface Web have much more flexible appearances, which makes the extraction a challenging task.

As we mentioned earlier, there are two types of information extraction techniques: Web information extraction and classic information extraction. Sine the homepages are normally not generated using templates, the traditional Web information extraction techniques are no longer suitable. The classic information extraction techniques can handle these

[2]http://libra.directtaps.net, username:guest, password:hello libra!



digc

Froogle Results **1** - **10** of about **12** confirmed / **17** total results for **digc**. (**0.22** seconds)

Sony DSC-P72 Cybershot Digital Camera 3.2M Pixel
$287.95 - Compare prices
SONY DSC-P72 CYBERSHOT DIGITAL CAMERA 3.2M PIXEL
ATACOM: 3.2 / 5

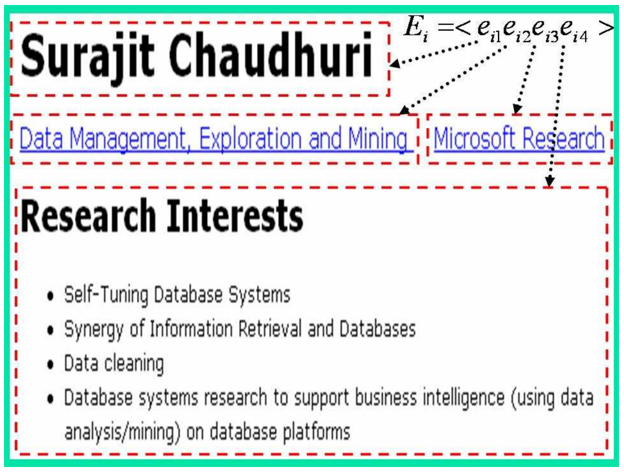
Sony DSC-F717 Digital Still Camera 5M Pixel
$698.95
SONY DSC-F717 DIGITAL STILL CAMERA 5M PIXEL
ATACOM: 3.2 / 5

Sony DSC-U60 Cybershot Digital Camera 2M Pixel
$318.95
SONY DSC-U60 CYBERSHOT DIGITAL CAMERA 2M PIXEL
ATACOM: 3.2 / 5

Sony DSC-V1 Cybershot Digital Camera Optical Zoom
$549.95 - Compare prices
SONY DSC-V1 CYBERSHOT DIGITAL CAMERA OPTICAL ZOOM
ATACOM: 3.2 / 5

Sony DSC-V1 Cybershot Digital Camera Optical Zoom
$549.95 - Compare prices
SONY DSC-V1 CYBERSHOT DIGITAL CAMERA OPTICAL ZOOM
ATACOM: 3.2 / 5

Figure 2: Four object blocks in a Web page



Surajit Chaudhuri

Data Management, Exploration and Mining Microsoft Research

Research Interests

- Self-Tuning Database Systems
- Synergy of Information Retrieval and Databases
- Data cleaning
- Database systems research to support business intelligence (using data analysis/mining) on database platforms

Figure 3: An example object block and its elements from a computer scientist home-page. Four object elements are located.

Web pages, however they consider solely plain text during the extraction, and ignore much valuable information, such as visual information of the Web pages and structured information from external databases, which can greatly assist the extraction process. For example, a foreigner could accurately locate the title and authors of the paper header in Figure 1 only depending on the font and position information. If a phrase in the paper header is found to exactly match with the *title* attribute of a record in the database, not only can this phrase be almost definitely labeled as *title*, but other attributes of this record, such as *author*, *affiliation* and *booktitle*, can be used to extract the rest information. Moreover, this fashion achieved a bi-directional communication between the extraction engine and the external databases, and a combined information extraction and integration (i.e. object identification).

2.2 Problem Formulation

Web Objects & Attributes: We define the concept of *Web Objects* as the principle data units about which Web information is to be collected, indexed and ranked. Web

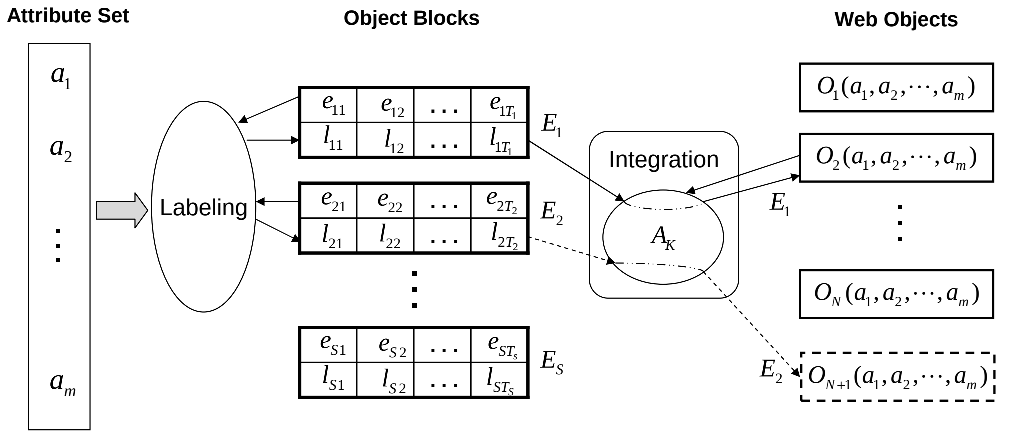

Attribute Set
Object Blocks
Web Objects
a_1
a_2
a_m
Labeling
Integration
A_K
E_1
E_2
E_S
$O_1(a_1, a_2, \cdots, a_m)$
$O_2(a_1, a_2, \cdots, a_m)$
$O_N(a_1, a_2, \cdots, a_m)$
$O_{N+1}(a_1, a_2, \cdots, a_m)$

Figure 4: Web Object Extraction

objects are usually recognizable concepts, such as authors, papers, conferences, or journals which have relevance to the application domain. Different types of objects are used to represent the information for different concepts. We assume the same type of objects follows a common relational schema: $R(a_1, a_2, ..., a_m)$.

Attributes, $A = \{a_1, a_2, ..., a_m\}$, are properties which describe the objects. There are three types of object attributes: Key attributes, Important attributes, and Others attribute:

- Key attributes, $A_K = \{a_{K1}, a_{K2}, ..., a_{KK}\} \subseteq A$, are properties which can uniquely identify an object;
- Important attributes, $A_I = \{a_{I1}, a_{I2}, ..., a_{II}\} \subseteq A$, are distinctive properties other than the key attributes;
- Others attribute, $a_O \in A$, is all the other properties about the object.

The designer of the system needs to determine the types of objects which are relevant to the application, and the Key and Important attributes of these objects.

Object Blocks & Elements: The information about an object on a Web page is usually grouped together as a block, since Web page creators are always trying to display semantically related information together. Using explicit or implicit visual separators such as lines, blank area, image, font, and color, we can first locate these object blocks based on existing Web page segmentation technologies like [9]. Figure 2 shows that four object blocks are located in a Web page generated by Froogle. With the help of data record mining [18] and classification [14] techniques, we can then automatically determine whether the object blocks are relevant to the application. However automated object block detection and classification is beyond the scope of this paper, where we assume that the relevant blocks are given.

Given an object block found on a Web page, it is straightforward to further segment it to atomic extraction entities using the visual information and delimiter, such as font, position, color, appearance pattern, and punctuation. Figure 3 shows an example object block with four atomic extraction entities, which are called *object elements*. In this way, the object block E_i is converted to a sequence of elements, i.e. $E_i =< e_{i1}e_{i2} \cdots e_{iT} >$. Each element e_{ij} only belongs to a single attribute of the object, and an attribute can contain several elements. See Section 3.2 for a detailed discussion on how to automatically obtain the object elements by block segmentation.

Web Object Extraction: Given an object block $E_i =< e_{i1}e_{i2} \cdots e_{iT} >$, and its relevant object schema $R(a_1, a2, ..., a_m)$, we need to assign an attribute name from the attribute set $A = \{a_1, a_2, ..., a_m\}$ to each object element e_{ij} to determine the corresponding label sequence $L_i =< l_{i1}l_{i2} \cdots l_{iT} >$. If the object block E_i and a previously extracted object O_n in the database refer to the same entity, we integrate O_n and the labeled E_i together. The key attributes A_K are used to decide whether they refer to the same entity. The combined labeling and integration inference is called Web object extraction.

Figure 4 shows such a Web object extraction process. Given an object block $E_i =< e_{i1}e_{i2} \cdots e_{iT} >$, we browse existing Web objects extracted previously during the labeling process to see if there are some good matches on key attributes between E_i and O_n. If we find such a match, it will be of high probability that they refer to the same object, and we can integrate their information together. For example, assuming E_1 and O_2 in Figure 4 match well on the key attributes A_K, we can use the information contained in E_1 to supplement or rectify the attributes of O_2. If there exist no good match, we build a new Web object whose attribute values are the labeled information of the object block. For example, assuming E_2 in Figure 4 has no match with existing Web objects on A_K, its labeled information is used to create a new Web object in the database.

2.3 Conditional Random Fields model

After locating an object block on Web pages and segmenting it to an object element set, the labeling operation can be treated as a sequence data classification problem. To the best of our knowledge, the Conditional Random Fields(CRF) model is among the most popular and effective methods for this task [16]. It offers several advantages over other Finite State Machine (FSM) based algorithms. First, it relaxes the strong independence assumptions made in those models. Second, it avoids a fundamental limitation, called "label bias" [16], of discriminative Markov models based on directed graphical models. Third, by designing appropriate feature functions, it is convenient to integrate arbitrary features into the model, which well meets our objectives. So, we select the CRF as the base model and extend it for Web object extraction.

Conditional Random Fields are undirected graphical models trained to maximize a conditional probability [16]. Usually a one-order linear chain is taken for simplicity. Specifically, given an object element (observation) sequence $E =< e_1 e_2 \cdots e_T >$ derived from an object block, a CRF models a conditional probability for a label (state) sequence $L =< l_1 l_2 \cdots l_T >$ as follows, where l_i belongs to a finite attribute alphabet $A = \{a_1, a_2, \cdots, a_m\}$,

$$P(L|E,\Theta) = \frac{1}{Z_E} \exp\left\{ \sum_{t=1}^{T} \sum_{k=1}^{N} \lambda_k f_k(l_{t-1}, l_t, E, t) \right\} \quad (1)$$

where, Z_E is the normalization constant that makes the probabilities of all possible label sequences sum to one.
$f_k(l_{t-1}, l_t, E, t)$ is called a feature function. It measures an arbitrary feature about the event that a transition $l_{t-1} \rightarrow l_t$ occurs at current time t and the total element sequence is E. $\Theta = \{\lambda_1, \lambda_2, \cdots, \lambda_N\}$ is the parameter set of the model, which indicates the relative importance of each feature function.

Using a CRF to deal with an labeling problem involves three phases of operations: model construction, model training, and output.

Model construction equals to the selection of the attribute set and feature functions. Usually feature functions are binary-valued, but they can also be real-valued as in this paper.

Model training is to determine the parameter set of the model. GIS [11], IIS [24], and L-BFGS [21] are normal training algorithms. Usually, L-BFGS is preferred due to its much faster convergence. More detailed comparisons among them can be found in [19, 31].

Output is to determine the optimal label sequence L^* with the highest probability, given an element sequence E and the learned CRF model:

$$L^* = \arg\max_{L} P(L|E,\Theta) \quad (2)$$

It can be efficiently tackled with the well-known Viterbi algorithm. Define $\delta_t(l)$ as the best score (highest probability) along a single path at time t , which accounts for the first t elements and ends in label l_t. By induction we have [25]:

$$\delta_t(l) = \max_{l'} \left\{ \delta_{t-1}(l') \exp\left[\sum_{k=1}^{N} \lambda_k f_k(l', l, E, t) \right] \right\} \quad (3)$$

After the recursion terminates at time T, we get

$$l^* = \arg\max_{l} [\delta_T(l)] \, ,$$

and the optimal state sequence L^* can be backtracked through the recorded dynamic programming table.

3. FEATURES FOR EXTRACTION

As we mentioned above, there exist three categories of information that could be utilized for Web object extraction: text features, visual features, and database features. In the following, we will discuss them respectively.

3.1 Text Features

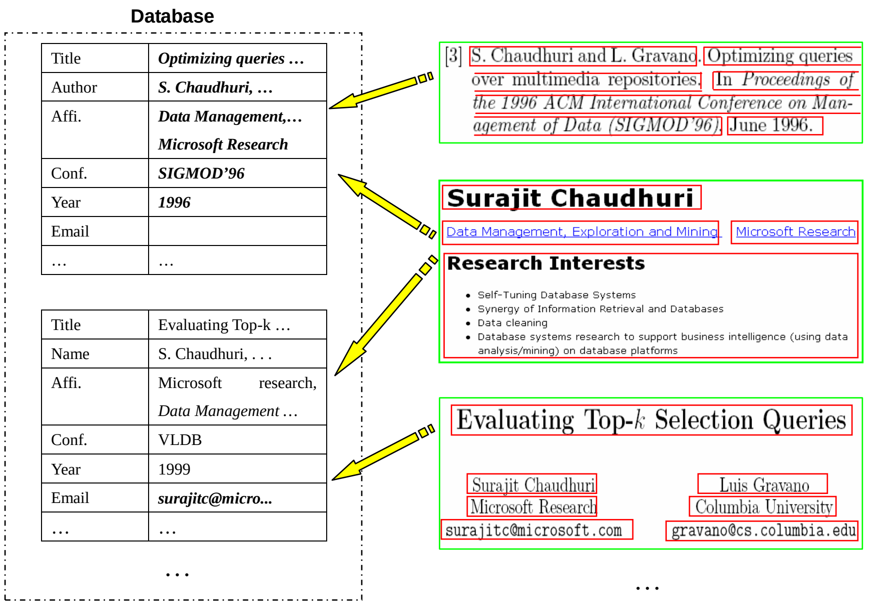
Text content is the most natural feature to use. Traditionally, the information of a Web object is treated as a sequence of words to be labelled. Statistics about word emission probabilities and state transition probabilities are computed on the training dataset, then these statistics are used to assist labelling the words one by one. But these word-by-word based approaches are not suitable for Web object extraction for the following reasons. First of all, although some high-level feature functions are very useful for the extraction, they are difficult to build based on single words due to their limited expression capability. For example, given "A. J. Black", we could say with high confidence that it is an *author* name. But little could be told based on individual word separately: "A.", "J.", and "Black". Given "Data", "Mining", we have no idea whether the labels should be *title* or *conference*, because they have similar emission probabilities for these two attributes. But if we treat "International Conference on Data Mining" as a whole, we could almost definitely say that labels of the five words are all *conference*. Secondly, because only one word's label is determined in one round, the labelling efficiency is impaired. Thirdly, usually it is straightforward to convert the information of an object block on the Web to an appropriate sequence of object elements, using visual features like font and position and delimiters like punctuation (refer to Subsection 2.1 for more details). Using Web elements as atomic entities could improve both the effectiveness and efficiency of the extraction engine.

The HTML tags of the Web pages are another type of text information which are widely utilized in traditional wrappers. But they are not so useful here because of their Website-dependent nature. Due to different designing styles among individual Website creators, information implied by tags is not stable.

3.2 Visual features

Object blocks usually contain many explicit or implicit visual separators such as lines, blank area, image, font size, and colors. They are very valuable for the extraction process. Specifically, it affects two aspects in our framework: block segmentation and feature function construction.

Using visual information together with delimiters is easy to segment an object block to an appropriate sequence of object elements. We prefer large elements under the condition that each element only corresponding to a single attribute. Having large elements has two advantages. First of all, more robust and powerful feature functions can be built because



Database
Title
Optimizing queries …
Author
S. Chaudhuri, …
Affi.
Data Management,…
Microsoft Research
Conf.
SIGMOD'96
Year
1996
Email
Title
Evaluating Top-k …
Name
S. Chaudhuri, . . .
Affi.
Microsoft research,
Data Management …
Conf.
VLDB
Year
1999
Email
surajitc@micro...
[3] S. Chaudhuri and L. Gravano. Optimizing queries over multimedia repositories. In Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD'96). June 1996.
Surajit Chaudhuri
Data Management, Exploration and Mining
Microsoft Research
Research Interests
Self-Tuning Database Systems
Synergy of Information Retrieval and Databases
Data cleaning
Database systems research to support business intelligence (using data analysis/mining) on database platforms
Evaluating Top-k Selection Queries
Surajit Chaudhuri
Microsoft Research
surajitc@microsoft.com
Luis Gravano
Columbia University
gravano@cs.columbia.edu

Figure 5: An example for Web object extraction. Attribute values in normal style are existing results in the database. Attribute values in italic style are currently obtained results by extracting from the left three object blocks.

the expression ability of an object element is generally in proportion to its length. Secondly, the extraction efficiency will be improved because more words can be labeled together in one round. For different kinds of Web objects, we should design particular element segmentation heuristic rules to determine the most appropriate element set. For example, for paper objects that contain visual information, like the paper header in Figure 5, , we can also use the font, position, and appearance patterns to obtain the element set. The final element set for the paper header is: { (Evaluating Top-k Selection Queries), (Surajit Chaudhuri), (Microsoft Research), (surajit@microsoft.com), (Luris Gravano), (Columbia University), (gravano@microsoft.com) }. Sometimes, regular expressions are incorporated to get a large element. For the author homepage in Figure 5, using the regular expression of "research interests" and visual appearances together, we can define the phrase of "research interests" together with the following four items as a single element. Achieving an appropriate object element set, high-level feature functions can then be built on them.

Visual information itself can also produce powerful features to assist the extraction. For example, if an element has the maximal font-size and centered at the top of a paper header, it will be the *title* with high probability. If two sub-blocks have similar patterns in appearance(for example, two authors' personal information in the paper header in Figure 5), the corresponding items in them should have the same labels. Though tag information is unstable across multiple heterogeneous Website, the visual information is much more robust, because people are always trying to display information on the web orderly and clearly, and this desirability makes the visual appearances of the same kind of objects vary much less than tags.

3.3 Database Features

For some kind of Web object, there normally exist some databases that contain large amounts of well-structured records which follow a similar relational schema with the object blocks to be processed. In our *Libra* context, DBLP and ACM DL are two of such databases. They store the essential attributes for thousands of papers, such as *title, author, booktitle* and *year*. This structured information can be used to remarkably improve the extraction accuracy in three ways.

First of all, we can treat the records in databases as additional training examples to compute the *element emission probability*, which is computed using a linear combination of the emission probability of each word within the element. In this way we can build more robust feature functions based on the element emission probabilities than those on the word emission probabilities.

Secondly, we can browse the databases to see if there are some matches between the current element and stored attributes, and apply the set of domain-independent string transformations to compute the matching degrees between them [30]. These matching degrees, which are normalized to the range of $[0 \sim 1]$, can be used to determine the label. For example, when extracting from the paper citation in Figure 5, its first element is "S. Chaudhuri and L. Gravano". It has a good match with the *author* attribute of the second record in the database. Then we can say with certain confidence that the label of the first element is *author*.

Thirdly, because the object blocks and the records in databases have the same relational schema, if we found good matches between some object elements and the key attributes of a record, we can say with high confidence that the object block and the record refer to the same object. Then we can use other attributes of this record to label the rest elements of the object block or rectify wrong labels. Take paper header in Figure 5 for an example. *Title* is a key attribute of a paper. For the first element, "Evaluating Top-k Selection Queries", we will find a good match with the *title* attribute of the second record in the database. It is of high probability that the header and the second record are about the same paper. Inversely, we browse the element sequence to see if there exist other matches with the rest attributes of the record. Then, the *name, affiliation* attributes will also find good matches. We compute the probability p_0 that the header and the record refer to the same object, and individual matching degrees p_i on each attribute using technologies of object identification [30]. Finally, we use all these matching results to direct further extraction or rectification.

We will show later that utilizing databases achieves an obvious improvement on extraction accuracy. Because the object identification operation is performed during the extraction process, information integration is also accomplished at the same time.

These three categories of features are all very valuable for Web object extraction. As far as we know, there is no framework which can efficiently integrate them together either in the field of Web IE and classical IE. In this paper, aiming at this goal, we extend the basic CRF to an Enhanced CRF model and propose an OLIE method for Web object extraction. The details will be formulated in the following section.

4. WEB OBJECT EXTRACTION

As stated above, our goal is to incorporate all available information to assist the Web object extraction. The basic CRF model can not meet this requirement, for it models the label sequence probability only conditioned on the element sequence $E =< e_1 e_2 \cdots e_T >$, and no object identification is performed. By introducing two variations into the basic CRF, we get the Enhanced CRF(ECRF) model, which well meets our objectives. In this section, we will first describe the ECRF model, then adopt it in the *Libra* context to execute the Web object extraction.

4.1 Enhanced Conditional Random Fields model

ECRF extends the basic CRF model by introducing two variations.

First, we modify the label sequence probability to condition on not only the element sequence, but also available databases,

$$P(L|E, D, \Theta) = \frac{1}{Z_E} \exp\left\{\sum_{t=1}^{T} \sum_{k=1}^{N} \lambda_k f_k(l_{t-1}, l_t, E, D, t)\right\} \quad (4)$$

where, E is the object element sequence, and it contains both the text and visual information. D denote databases which store structured information. $f_k(l_{t-1}, l_t, E, D, t)$ is the new feature function based on all the three categories of information.

Sometimes we would have sufficiently high confidence that some object element e_t should have certain label. The cases may be that good matches between e_t and key/important attributes of records in databases are found, or that e_t has a high enough element emission probability for some attribute. For example, if the following statistics holds, $p(l_t = \text{"}conference\text{"}|e_t$ contains "in proceedings of"$) = 0.99$, and current e_t is "in proceedings of SIGMOD04", it is almost definite that *conference* is the label. These constraints can be used to guide the solution searching progress to find the optimal label path correctly and quickly. This leads to our second variation for the basic CRF. Specifically, we first compute the confidence $c_t(a_i)$ that e_t belongs to certain attribute a_i based on some feature functions. If the confidence is high enough($c_t(a_i) > \tau$), we modify the induction formula of Viterbi algorithm as follows,

$$\delta_t(l) = \begin{cases} \max\limits_{l'} \left\{ c_t(a_i) \cdot \delta_{t-1}(l') \exp\left[\sum\limits_{k=1}^{N} \lambda_k f_k(l', l, E, D, t)\right]\right\} & l = a_i \\ \max\limits_{l'} \left\{ (1 - c_t(a_i)) \cdot \delta_{t-1}(l') \exp\left[\sum\limits_{k=1}^{N} \lambda_k f_k(l', l, E, D, t)\right]\right\} & others \end{cases} \quad (5)$$

if $c_t(a_i) \leq \tau$, the induction formula is the same as (3).

The above two variations further enhanced the ability of the basic CRF. It makes it possible to incorporate more valuable information during the extraction process. Moreover, the modified Viterbi algorithm leads to a more efficient solution searching progress.

4.2 Object Level Information Extraction

In this subsectin, we will take the *Libra* context for example, and adopt the ECRF model to achieve the OLIE approach for Web object extraction.

We need to process two kinds of Web objects: papers and authors. (currently only text-related information is extracted. Others like image are not considered.) Text, visual, and database features are all the available information during the extraction process.

For each kind of Web objects, we should construct a particular ECRF model for it. This involves two operations: determining the attribute set and constructing appropriate feature functions. Then, the parameter set of the ECRF model is learned on a training dataset.

Given an object block to be processed, we first convert it to a sequence of elements $E =< e_1 e_2 \cdots e_T >$, according to the visual features and delimiters. Then we determine the corresponding label sequence with the modified Viterbi algorithm. Specifically, at time $t = 1 : T$, the following operations are performed in succession:

1. compute each feature function $f_k(l_{t-1}, l_t, E, D, t)$.

2. Find constraints:

(1) If the element emission probability s_i of e_t as certain attribute a_i is high enough($s_i \geq \tau$), where τ is a preset threshold, we take s_i as the constraint confidence $c_t(a_i)$ and record a tuple $< e_t, a_i, c_t(a_i) >$.

(2) If e_t finds a good match with the attribute a_i of a record in the database, we take the normalized matching degree m_i(in the range of $[0 \sim 1]$) as the confidence $c_t(a_i)$. If

```
OLIE Algorithm
  1. Get object element sequence $E =< e_1 e_2 \cdots e_T >$ by
     visual features and delimiters.
  2. for $t = 1 : T$
     A. Compute $f_k(l_{t-1}, l_t, E, D, t), k = 1 : N$.
     B. Find constraints:
        (a). If element emission probability $s_i \geq \tau$,
             take $c_t(a_i) = s_i$ and record the tuple
             $< e_t, a_i, c_t(a_i) >$.
        (b). If attribute matching degree $m_i \geq \tau$, take
             $c_t(a_i) = m_i$ and record the tuple
             $< e_t, a_i, c_t(a_i) >$.
        (c). If object matching degree $p_0 \geq \tau$,
             inversely browse $E$ to compute each
             attribute matching degree $p_j$. If $p_j \geq \tau$,
             take $c_{t'}(a_j) = max(p_j, p_0)$ and record a
             tuple $< e_{t'}, a_j, c_{t'}(a_j) >$, where
             $t' \in \{1, 2, \cdots, T\}$.
     C. For all $t' \leq t$, pick up the tuple
        $< e_{t'}, a_i^*, c_{t'}(a_i)^* >$ with the highest
        confidence. From the smallest time $t'$ to $t$ ,
        reperform the searching process with the
        modified Viterbi algorithm.
  3. Terminate at $l^* = \arg\max_l [\delta_T(l)]$ and backtrack
     the optimal state sequence.
```

Figure 6: The OLIE algorithm

$c_t(a_i) \geq \tau$, we also record the tuple $< e_t, a_i, c_t(a_i) >$.

(3) If the matched attribute is a key attribute of the record, we inversely browse E to compute the normalized matching degree p_j for each attribute a_j of the record, and compute the probability p_0 that the current object block and the record refer to the same object. For each attribute a_j, if $p_j \geq \tau$ holds, we take the bigger of p_j and p_0 as the confidence $c_{t'}(a_j)$, and record a tuple $< e_{t'}, a_j, c_{t'}(a_j) >$, where $t' \in \{1, 2, \cdots, T\}$.

3. For all $t' \leq t$, we pick up the tuple $< o_{t'}, l_i^*, c_{t'}(a_i)^* >$ with the highest confidence. From the smallest time t' to the current time t , we re-perform the searching process with the modified Viterbi algorithm.

4. When time T is reached, we terminate at $l^* = \arg\max_l [\delta_T(l)]$ and backtrack the optimal state sequence.

Based on ECRF, our OLIE sufficiently utilizes all available information to assist the extraction for Web objects. Because object identification is performed during this process, a bidirectional communication among object blocks and records of databases is achieved, which leads to a combined information extraction and integration.

5. EXPERIMENTS

The OLIE approach proposed in the paper are fully implemented and evaluated in the context of *Libra*. The goals of the experimental study are: (i) to compare the performance of our OLIE approach with that of existing classic IE approaches, (ii) to see how different sizes of the external databases could affect the extraction accuracy;

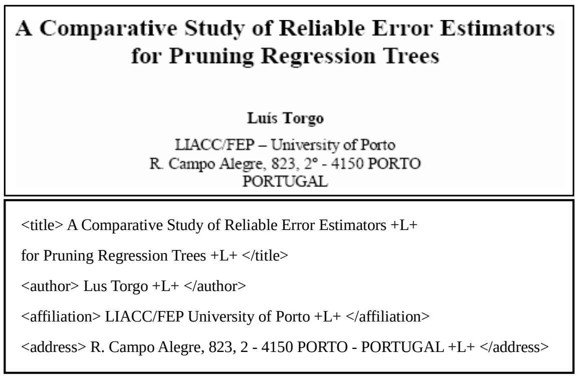
5.1 Experimental setup

Two types of Web objects are defined in the experiments: papers and authors. The paper objects have 8 attributes: Title, Author, Year, Abstract, Editor, Booktitle, Journal, and Others, and the key attributes are: Title, Author, and Year. The author objects have 13 attributes: Name, Degree, Affiliation, Designation, Address, Email, Phone, Fax, Education, Secretary, Office, Web URL, and Others, and the key attribute is the Name of the author.

We now describe the datasets and metrics of our experimental evaluation. We also discuss the feature functions used in the experiments.

5.1.1 Datasets

Citations: We took the citation dataset derived from the Cora project for testing. It has been used as a standard benchmark in several previous works [13, 23, 27]. It contains 500 citations and we used 300 for training and the rest 200 for testing. 7 attributes of paper objects are extracted: Author, Title, Editor, Booktitle, Journal, Year, and Others. This data set is denoted as **C**.



A Comparative Study of Reliable Error Estimators for Pruning Regression Trees

Luis Torgo

LIACC/FEP – University of Porto
R. Campo Alegre, 823, 2° - 4150 PORTO
PORTUGAL

<title> A Comparative Study of Reliable Error Estimators +L+
for Pruning Regression Trees +L+ </title>
<author> Lus Torgo +L+ </author>
<affiliation> LIACC/FEP University of Porto +L+ </affiliation>
<address> R. Campo Alegre, 823, 2 - 4150 PORTO - PORTUGAL +L+ </address>

Figure 7: Header examples from web and Cora dataset

PDF Files: Though a header dataset is also provided by the Cora project, it is not appropriate for our testing because it only contains the plain text of a header, discarding much visual information like font and position. This makes its appearance remarkably different from the original one in a PDF/PS science paper (see Figure 7 for an example). To test our algorithm's performance, we need the visual information of the original PDF/PS files. we randomly selected 200 papers in the dataset and downloaded them from the internet. Their original headers composed a dataset denoted as **H**. We used 100 papers for training and the rest for testing. 9 attributes of author objects are extracted: Name, Affiliation, Address, Email, Fax, Phone, Web URL, Degree, and Others. 4 attributes of paper objects are extracted: Title, Author, Abstract, and Others.

Table 1: List of some used features

Text	All words start with capitalized letter
	Initial word starts with capitalized letter
	Form like author name, such as *S. Chaudhuri*
	Number of digits in the element
	Percent of digits in the element
	Number of words in the element
	Element emission probability
	Contain at least one -
	Phone number or zip code
	Regular expression, such as URL, email
Vision	Font information like size, family and style
	Position like top, left and right
	Distance from previous element
	Distance from next element
	Same font with previous element
	Same font with next element
Database	Match an attribute of a record
	Match a key attribute of a record
	Contain word like *Jan., Feb.*
	Contain phrase like *submitted to, etc*
	Contain phrase like *in proceedings of, etc*

Author homepages: We randomly collected 200 computer scientists' homepages from the internet. Compared with previous two datasets, this dataset is more general and flexible. 11 attributes of the author objects are extracted: Name, Affiliation, Designation, Address, Email, Phone, Fax, Education, Secretary, Office, and Others. We randomly selected 100 homepages for training and the rest 100 for testing. This dataset is denoted as **P**.

ACM Digital Library: ACM Digital Library is online Web database with high quality structured data, which totally contains essential structured information about 150,000 papers on computer science.

5.1.2 Evaluation criteria

To evaluate the performance comprehensively, we took several criteria that were widely used before: word accuracy, F1-measure and instance accuracy. They measure the performance of the algorithms on different aspects. A brief definition is given as follows [23].

- Word accuracy: Defining A as the number of true positive words, B as the number of false negative words, C as the number of false positive words, D as the number of true negative words, $A+B+C+D$ is the total number of words. Word accuracy is calculated to be
$$acc. = \frac{A+D}{A+B+C+D}$$
- F1-measure: First, two basic measurements, precision and recall, are defined as: $Precision = \frac{A}{A+C}, Recall = \frac{A}{A+B}$. Then F1-measure is calculated to be
$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$
- Instance accuracy: the percentage of instances in which all words are correctly labelled.

5.1.3 Feature functions

During our experiments, feature functions are constructed based on the three kinds of available features described in section 3: text, vision, and database. And they are all based on object elements instead of words. A more detailed description is listed in Table 1. For the paper citation dataset **C**, the visual information (punctuation) is used to get object elements, and feature functions are only based on the text and database features. For paper header dataset **H** and author homepage dataset **P**, feature functions are based on all the three kinds of features.

5.2 Experimental Results

We now discuss the experimental results on the three datasets **H**, **C**, **P**. For **H** and **C**, we also made a comparison with several typical algorithms, such as HMM[27], SVM[13], and CRF[23]. Note that the structured information from the ACM DL is used during the extraction process. Because the definitions of attributes are not exactly the same, we only compared the extraction results on some common ones.

Table 2, Table 3, and Table 4 list the *word accuracy* and *F1-measure* on each field, and Figure 8 shows the *instance accuracy* on three datasets. From the reports in Table 2 and Table 3, we can see that OLIE achieves an obvious, sometimes dramatic, improvement on almost each field compared with existing methods. It substantiates that in our OLIE framework, all available information about web objects is more sufficiently utilized during the extraction process, and this mode does help a lot to improve the extraction performance. Particularly, we nearly obtain a perfect extraction result on the paper header dataset H. The reason is that, visual information in a paper header is especially abundant and stable. It enable us to build powerful feature functions to assist the extraction.

Though author homepage is more difficult to deal with due to its more flexible appearance, we still obtain a satisfying extraction result as shown in Table 4.

For a user, usually he/she cares more about whether an object block is correctly labeled than how many words of the obejct are correctly labeled. So the instance accuracy is a more practical criterion. In Figure 8, we show OLIE's extraction results on this criterion and compared them with some typical algorithms on C and H. An obvious improvement is obtained due to two main reasons. First, additional information such as vision and database is utilized to help the extraction. Second, the labeling process is based on elements instead of words.

To test the effectiveness of using object elements instead of words, we discard database features during the extraction on C. The result is shown in Figure 8 corresponding to the OLIE* method. We can see that, though the result is not so satisfying as OLIE, an improvement is still obtained compared with CRF and HMM.

To test the effectiveness of utilizing database information, we vary the size of database during the extraction on C.

Table 2: Extraction results on C

	HMM		CRF		ECRF	
	acc.	F1	acc.	F1	acc.	F1
Author	96.8	92.7	99.9	99.4	99.9	99.9
Title	92.2	87.2	98.9	98.3	99.5	99.1
Booktitle	94.4	0.85	97.7	93.7	99.2	97.2
Journal	96.6	67.7	99.1	91.3	99.6	96.6
Editor	98.8	70.8	99.5	87.7	99.8	95.2
Date	99.7	96.9	99.8	98.9	100	100

Table 3: Extraction results on H

	HMM		SVM		CRF		ECRF	
	acc.	F1	acc.	F1	acc.	F1	acc.	F1
Author	98.7	81.0	99.3	97.2	99.8	97.5	100	100
Title	98.2	82.2	98.9	96.5	99.7	97.1	100	100
Affiliation	98.3	85.1	98.1	93.8	99.7	97.0	99.9	99.9
Address	99.1	84.8	99.1	94.7	99.7	95.8	99.9	99.8
Email	99.9	92.5	99.6	91.7	99.9	95.3	100	100
Phone	99.8	53.8	99.9	92.4	99.9	97.9	100	100
Web	99.9	68.6	99.9	92.4	99.9	94.1	100	100
Degree	99.5	68.6	99.5	70.1	99.8	84.9	100	100

Table 4: Extraction results on P

	ECRF	
	acc.	F1
Name	99.9	99.4
Affiliation	99.6	99.2
Designation	99.8	99.0
Address	98.3	95.2
Email	99.9	99.6
Phone	99.2	95.8
Education	99.9	99.9

Table 5: Extraction results on C with databases of various sizes.

Paper Num	0		5000		30000		150000	
	acc.	F1	acc.	F1	acc.	F1	acc.	F1
Author	99.31	98.53	99.82	99.61	99.95	99.89	99.97	99.94
Title	98.61	97.69	99.29	98.84	99.41	99.03	99.46	99.12
Booktitle	98.79	96.01	99.03	96.83	99.03	96.83	99.15	97.23
Journal	99.38	94.03	99.39	94.12	99.40	94.37	99.64	96.59
Editor	99.69	93.94	99.69	93.94	99.73	94.81	99.75	95.15
Date	99.54	93.46	99.62	94.56	99.63	94.79	100	100

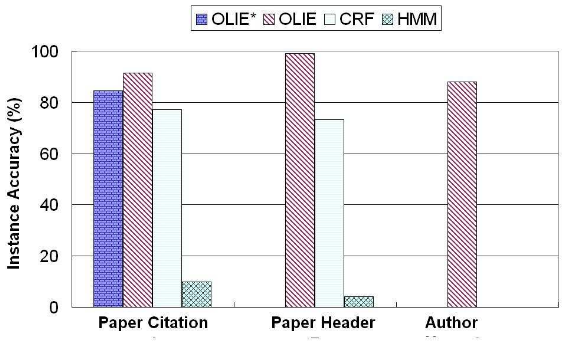


Figure 8: Instance accuracy by different algorithms

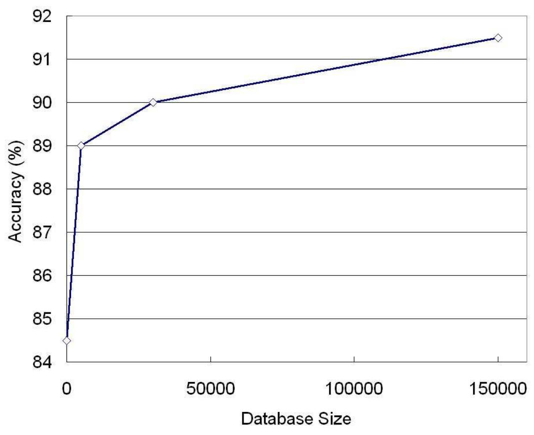


Figure 9: Instance accuracy V.S. database size on C

Specifically, we randomly selected 0, 5000, 30000, and 150000 papers from ACM DL to derive different databases, and conducted individual experiment on each of them. The instance accuracy is shown in Figure 9. When we increase the database size, we obtain a gradual improvement on accuracy. More interesting, the slope of the accuracy curve becomes flatter and flatter as we increase the database size. It shows that there exists a distinction between whether using a database or not during the extraction process. But with more and more database information incorporated, its effectiveness will degrade gradually. This hint is important for us to design a practical system. Using database information during the extraction process will involve many object identification operations. When the size of database is too large, it will be very time-consuming. Given the accuracy curve in Figure 5, it enable us to select a medium-sized database to find a good trade-off between extraction accuracy and efficiency.

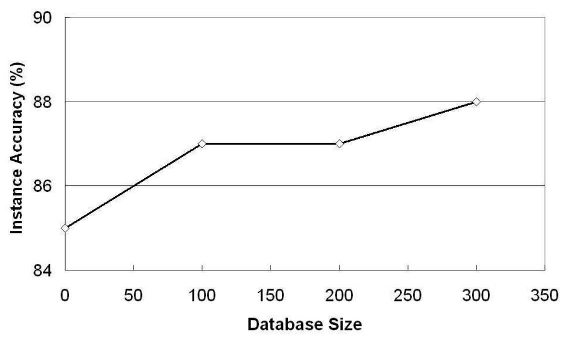


Figure 10: Instance accuracy for different database sizes

Sometimes, we would have no external databases to assist the extraction at the beginning for some certain types of Web objects. With the extraction going on, we can build a local database using the extracted results. To test the effectiveness of utilizing this kind of database, we designed a simple experiment on c. Specifically, we randomly selected 100 citations from C as training examples to construct the ECRF model and another 100 citations as test examples. Then we randomly selected 100 instances from the remaining 300 citations for extraction and built a local database composed of the extracted results. We denoted it as D_1. Further, we randomly selected another 100 instances from the remaining 200 citations for extraction and add the extracted results to D_1. The new database is denoted as D_2. For the last 100 citations, we re-performed the extraction process and also added the extracted results to D_2 and obtained D_3. Then, we evaluated the extraction accuracy for the previously selected 100 testing examples by using databases D_1, D_2, and D_3 respectively. The results are shown in Figure 10. We also achieved an obvious improvement by using D_1. The improvement rate slowed down at D_2 and D_3. This preliminary experiment shows that it is possible to obtain a self-enhanced Web object extraction system using our OLIE approach: on one hand, the growing database can be very helpful in extracting more accurate results. On the other hand, the extracted results which are more accurate can further increase the size and quality of database.

6. RELATED WORK

As stated above, the proposed OLIE framework is based on the ECRF model, a Finite State Machine(FSM) model. There has been much research on this topic recently [16, 20, 27, 28, 35]. Introducing FSM into the information extraction field achieved an obvious improvement on extraction accuracy compared with previous rule-based extraction methods. But little information on vision and database was used during previous works, and documents to be extracted were treated as sequences of words instead of object elements.

Sunita Sarawagi and William W. Cohen propose to sequentially classify segments of several adjacent words in name entity extraction [26]. They formalized a semi-Markov Conditional Random Fields model and used it for segmenting and labeling plain text documents. Differently, our OLIE is based on the ECRF model, and directly uses visual features to obtain the sequence of object elements from Web pages. It is more efficient for Web objects extraction.

A number of other techniques have been proposed to use external dictionaries during information extraction [7, 23, 27, 10]. Generally, These dictionaries were used in two ways. First of all, they were used as training examples to compute emission probabilities. Second, a matching degree among the current observation and entities in dictionaries was computed to help determining the label of the current observation. In our OLIE approach, databases are utilized instead of dictionaries. In addition to the above two possible usages of dictionaries, an external database could be much more helpful. A dictionary is always about a single attribute, while records in a database normally have multiple attributes which follow a relational schema. When the current object element finds a good match with the key attribute of a record in the database, the rest attributes of this record can be used to determine or rectify labels of other elements.

Due to the introduction of databases into the extraction process, we often have high enough confidence that some elements should have certain labels. We modified the Viterbi algorithm to utilize this constraint to compel the solution searching progress to find the optimal state path correctly and quickly. This is very similar with the idea in [29], where a constrained Viterbi algorithm was proposed. But there still exists some difference. In [29], the extraction results with the lowest confidence would be sent to people for checking. Then constrained Viterbi algorithm is performed to relabel the sequence of words according to people's rectifications. So the constraints were introduced manually. In our OLIE approach, the probabilities that some elements should have certain labels are automatically computed through the interaction with databases. Since we only have some confidence that the labels are correct, we need to utilize the constraints in a probabilistic way.

7. CONCLUSION AND FUTURE WORK

How to accurately extract structured data from the Web has led to significant interest recently. Many valuable researches have been conducted on this area. However, as for Web object extraction, which targets to extract and integrate all the related Web information about the same object together as an information unit, there is still no ready solution. By leveraging the advantages of both Web IE and classic IE techniques, we propose an *Object-Level Information Extraction (OLIE)* approach by extending the *Conditional Random Fields (CRF)* algorithm with more Web-specific information such as vision features and database features. The novelty of this approach lies in that it utilizes as much available Web information as possible to assist the extraction process. We adopt the proposed OLIE method in the *Libra* context, a science paper search engine. Experimental results on our datasets show that OLIE always outperforms previous baselines, sometimes dramatically.

In the next step, we plan to further imporve the accuracy of identifying object blocks and object elements from Web pages, which is a big factor to affect the final extraction accuracy. The good news is that, according to our current experience, high accuracy can be expected for them since Web page authors rarely encode the objects randomly, especially for those high-quality websites. We also prepare to apply the OLIE method to other domains, such as product object extraction, which is a good way to test the generality of this method.

8. REFERENCES

[1] Libra. Object-level Paper Search Engine. http://libra.directtaps.net, username:guest, password:hello libra!

[2] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *ACM SIGMOD Conference (SIGMOD)*, 2004.

[3] L. Arlotta, V. Crescenzi, G. Mecca, and P. Merialdo. Automatic annotation of data extracted from large web sites. In *International Workshop on the Web and*

Databases (WebDB), 2003.

[4] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. In *Proc. Workshop on Management of Semistructured Data*, Tucson, 1997.

[5] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a high-performance learning name-finder. In *Proceedings of ANLP*, 1997.

[6] M. Bilenko, R. Mooney, W. W. Cohen, P. Ravikumar, and S. Fienberg. Adaptive name-matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[7] V. R. Borkar, K. Deshmukh, and S. Sarawagi. Automatic text segmentation for extracting structured records. In *International Conf. on Management of Data (SIGMOD)*, 2001.

[8] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[9] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma. Block-based web search. In *ACM SIGIR Conference (SIGIR)*, 2004.

[10] W. Cohen and S. Sarawagi. Exploiting dictionaries in named entity extraction: Combing semi-markov extraction processes and data integration methods. In *KDD*, 2004.

[11] J. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *the Annals of Mathematical Statistics*, 43(5), 1972.

[12] C. Giles, K. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In *ACM Conference on Digital Libraries*, 1998.

[13] H. Han, C. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. Fox. Automatic document metadata extraction using support vector machines. In *ACM/IEEE Joint Conference on Digital Libraries(JCDL)*, 2003.

[14] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmman Publishers, 2000.

[15] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper induction for information extraction. In *Intl. Joint Conference on Artificial Intelligence (IJCAI)*, pages 729–737, 1997.

[16] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[17] K. Lerman, L. Getoor, S. Minton, and C. A. Knoblock. Using the structure of web sites for automatic segmentation of tables. In *ACM SIGMOD Conference (SIGMOD)*, 2004.

[18] B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.

[19] R. Malouf. A comparison of algorithms for maximum entropy parameter estimaiton.

[20] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML*, 2000.

[21] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.

[22] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[23] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*, 2004.

[24] S. Pietra, C. Pietra, and J. Lafferty. Inducing features of random fields. *IEEE PAMI*, 19(4), 1997.

[25] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*, 77.

[26] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In *NIPS*, 2004.

[27] K. Seymore, A. McCallum, and R. Rosenfeld. Learning hidden markov model structure for information extraction. In *AAAI Workshop*, 1999.

[28] M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden markov model for information extraction. 2003.

[29] P. V. A. M. T. Kristjansson, A. Culotta. Interactive information extraction with constrained conditional random fields.

[30] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Knowledge Discovery and Data Mining (KDD)*, 2002.

[31] H. Wallach. Efficient training of conditional random fields, 2002. master thesis.

[32] J. Wang and F. H. Lochovsky. Data extraction and label assignment for web databases. In *World Wide Web conference (WWW)*, 2003.

[33] J. Wang, J.-R. Wen, F. H. Lochovsky, and W.-Y. Ma. Instance-based schema matching for web databases by domain-specific query probing. In *Very Large Data Bases (VLDB)*, 2004.

[34] Y. Wang and J. Hu. A machine learning based approach for table detection on the web. In *World Wide Web conference (WWW)*, 2002.

[35] B. Wellner, A. McCallum, and F. Peng. An integrated, conditional model of information extraction and coreference with application to citation matching. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004.

[36] S. Yu, D. Cai, J. rong Wen, and W. ying Ma. Improving pseudo-relevance feedback in web information retrieval using web page segmentation. In *World Wide Web Conference Series*, 2003.