Towards Combining Web Classification and Web Information Extraction: a Case Study

Ping Luo, Fen Lin, Yuhong Xiong, Yong Zhao, Zhongzhi Shi

HP Laboratories
HPL-2009-86

Keyword(s):
Classification, Information extraction, Graphical model

Abstract:

Web content analysis often has two sequential and separate steps: Web Classification to identify the target Web pages and Web Information Extraction to extract the metadata contained in the target Web pages. This decoupled strategy is highly ineffective since the errors in Web classification will be propagated to Web information extraction and eventually accumulate to a high level. In this paper we study the mutual dependencies between these two steps and propose to combine them by using a model of Conditional Random Fields (CRFs). This model can be used to simultaneously recognize the target Web pages and extract the corresponding metadata. Systematic experiments in our project OfCourse for online course search show that this model significantly improves the F1 value for both of the two steps. We believe that our model can be easily generalized to many Web applications.

External Posting Date: April 21, 2009 [Fulltext] Approved for External Publication
Internal Posting Date: April 21, 2009 [Fulltext]

To be published and presented at the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Towards Combining Web Classification and Web Information Extraction: a Case Study

Ping Luo[†], Fen Lin[‡], Yuhong Xiong[†], Yong Zhao[†], Zhongzhi Shi[‡]
[†]Hewlett Packard Labs China, {ping.luo, yong.zhao, yuhong.xiong}@hp.com
[‡]Institute of Computing Technology, CAS, {linf, shizz}@ics.ict.ac.cn

ABSTRACT

Web content analysis often has two sequential and separate steps: Web Classification to identify the target Web pages, and Web Information Extraction to extract the metadata contained in the target Web pages. This decoupled strategy is highly ineffective since the errors in Web classification will be propagated to Web information extraction and eventually accumulate to a high level. In this paper we study the mutual dependencies between these two steps and propose to combine them by using a model of Conditional Random Fields (CRFs). This model can be used to simultaneously recognize the target Web pages and extract the corresponding metadata. Systematic experiments in our project *OfCourse* for online course search show that this model significantly improves the F1 value for both of the two steps. We believe that our model can be easily generalized to many Web applications.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical

General Terms

Algorithms, Experimentation

Keywords

Classification, Information extraction, Graphical model

1. INTRODUCTION

The explosive growth and popularity of the World Wide Web has resulted in a huge amount of information on the Internet. Currently access to this huge collection of information has been limited to searching and browsing at the level of Web pages. However, sophisticated Web applications, such as vertical search and entity search, call for flexible information extraction techniques that transform information on the Web pages into structured (relational) data [7]. In recent years, progress in this area has moved us closer to this goal. Some research and industry groups have built systems that support more precise query on certain content verticals, such as online products [2], academic publications [8] etc.

Usually, to build such systems we need a process that is a cascade of two main steps, i.e. *Web classification* and *Web information extraction*, after the relevant Web pages are downloaded by *focused crawling* [10]. Since focused crawling inevitably includes some irrelevant pages we need to conduct a finer level of Web-page classification to separate out the relevant pages from the irrelevant ones. Then, after the Web pages are further categorized Web information extraction is performed on the *real* relevant pages to extract target entities with their metadata. As stated by McCallum [7], it is highly ineffective to use this decoupled strategy - attempting to do Web classification and Web information extraction in two separate phases. Specifically, since these two steps are separate, the errors in Web classification will be propagated to Web information extraction and eventually accumulate to a high level. Therefore, the overall performance is upper-bounded by that of Web classification.

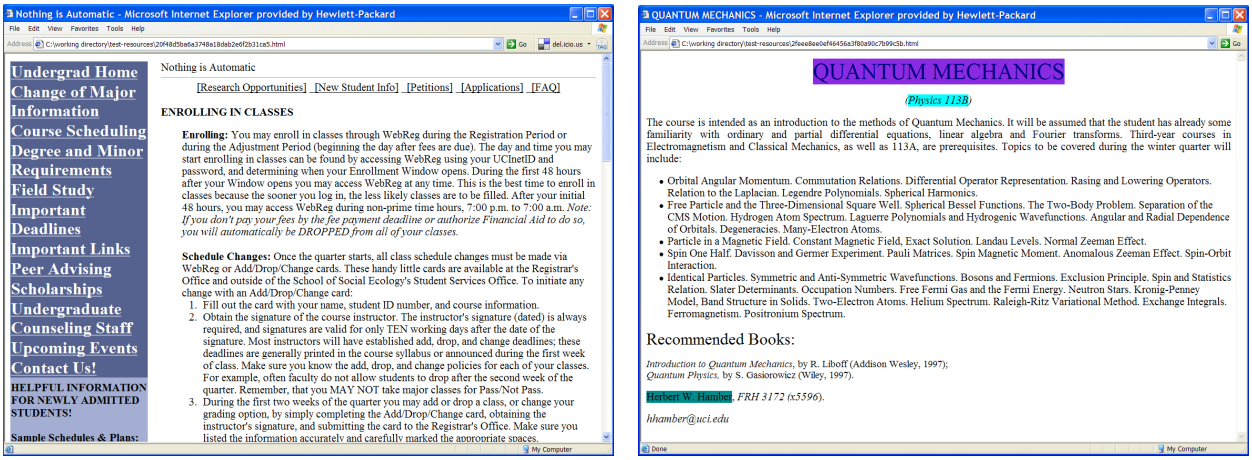
In this paper, we propose a method to combine Web classification and information extraction and achieve mutual enhancement between these two operations. Rather than conducting these two steps separately and sequentially, our method utilizes the probabilistic graphical model to simultaneously detect the target Web pages and extract the metadata in them, through which we aim to improve both the recall and precision of Web classification and Web information extraction.

1.1 Motivating Examples

We begin by illustrating the problem with an example, drawn from a project *OfCourse* at HP Labs China. *OfCourse*[1] is a vertical search engine for online courses. The goal of this project is to accurately identify the online course pages and extract key information from them. The key information include course title, course time, course ID and so on, and we call them course metadata. The traditional approach to reach this goal is to first identify the course homepages by Web page classification, then extract metadata from the homepages. After studying this problem, we find that these two steps are closely related in that information obtained from one step can greatly help the other step. On one hand, if a Web page is a course homepage it usually contains a course title. On the other hand, the existence of some course metadata, in turn, indicates that the current

[1]Available at http://fusion.hpl.hp.com/OfCourse/

KDD'09, June 28–July 1, 2009, Paris, France.



(a) with course features and without course metadata

(b) without course features and with course metadata

Figure 1: The Motivating Examples of Course Homepages

Web page is a course homepage. This means that there is a forward dependency from Web classification to information extraction, and also a backward dependency from Web information extraction.

To understand the room for improvement by considering the backward dependency at an intuitive level, we give two typical course homepages, as shown in Figure 1. Before we detail these two examples we first give a brief introduction on how to identify the course homepages by text classification. The online course homepages usually contain some terms about course and teaching, e.g., "*instructor*", "*teaching assistant*", "*course scheduling*", "*syllabus*" etc. These terms, which frequently appear in course homepages, can be used as features to distinguish them from non course homepages. However, the occurrence of these terms is not a sufficient condition for course homepages. The Web page in Figure 1(a) describes how to enroll a class, and also contains many terms about course and teaching. Thus, it is likely that a classifier makes a wrong decision by predicting it to be a course homepage. As to the course homepage in Figure 1(b), it contains a conspicuous course title "*Quantum Mechanics*" with the course ID "*physics 113B*". However, it lacks the course terms to indicate it is a course homepage ("*Recommended Books*" is the only course term it contains). Thus, it is likely that a classifier predicts the page in Figure 1(b) to be not a course homepage. To show the effect of the dependency from information extraction to classification, we assume for the moment that there exists an *oracle* to tell whether a Web page contains a course title or not, and a Web page is a course homepage if and only if it contains a course title. Then, under this assumption we can reach the right conclusion that the Web page in Figure 1(a) is not a course homepage since it does not contain a course title, and the one in Figure 1(b) is a course homepage since it contains a course title. These two examples show that we can leverage the existence of certain entity metadata to improve not only the precision (shown by the example in Figure 1(a)) and but also recall (shown by the example in Figure 1(b)) in classification and extraction. However, how can we obtain the oracle for course metadata? This is the problem we address in this paper.

1.2 Our Solution

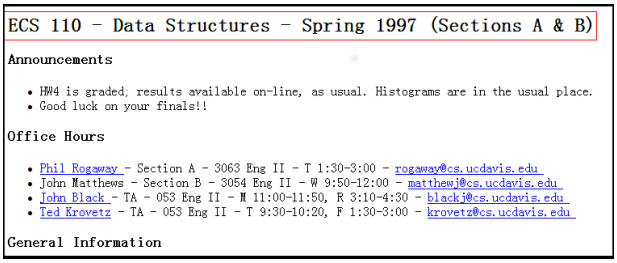
As described above, we find that Web classification and Web information extraction are actually two coupled steps, and they should not be separated. Therefore, in this paper we propose a method to jointly and simultaneously optimize these two steps by the probabilistic graphical model CRFs [5].

Given a Web page x and a set of DOM (Document Object Model) tree leaf nodes $x^1, \cdots, x^k$ in x, we formulate the problem of Web classification and Web information extraction as the task of assigning labels to $x, x^1, \cdots, x^k$. The label on x indicates the class of this Web page, while the labels on $x^1, \cdots, x^k$ indicates the types of the metadata for each leaf node. This way these two tasks can be integrated into one probabilistic model. Additionally, we explicitly consider the constraints between Web classification and Web information extraction (actually the constraints among the labels of $x, x^1, \cdots, x^k$), and inject these constraints in both model learning and prediction. This way we greatly reduce the label space and achieve *exact inference* efficiently.

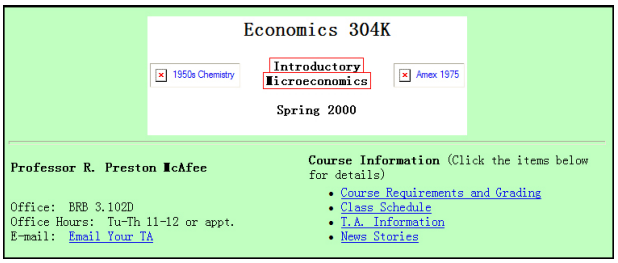
To the best of our knowledge, our approach is the first to simultaneously conduct Web classification and Web information extraction by a unified graphical model. Specifically, we make the following contributions: 1) A mutually beneficial integration of Web classification and Web information extraction. This method explicitly consider both the forward and backward dependencies between Web classification and Web information extraction. It shows significant improvement on the F1 value of these two steps. 2) An empirical study of this combination method on the task of online course search. Two baseline methods are proposed and compared with our uniform graphical model. 3) The online course search engine *OfCourse*. It is built based on the method proposed in this paper and it is fully functional for public access now. It currently contains over 60, 000 courses from the top 50 schools in the US.

2. PROBLEM FORMULATION

In the following we use our work in online course search as an example to formulate the problem of combining Web classification and Web information extraction.



(a) Course homepages 1



(b) Course homepages 2

Figure 2: Examples of course homepages

2.1 Problem Description

In this work Web classification is to identify the course homepages in a given group of Web pages and Web information extraction is to extract the course metadata from the course homepages. Here, the course homepages, whose contents are provided by the teachers of the courses, include all the information about the course. Figure 2 gives two example segments of course homepages. In this paper, we consider three types of course metadata:

- *Course Title:* the name of the course.
- *Course ID:* the identifier of the course, often provided by the university or the department.
- *Course Time:* the semester or time of the year for the course (it is not the time of the week when the lectures are offered, like Tuesday and Thursday).

These three kinds of course metadata in Figure 2(a) are "Data Structures", "ECS 110", and "Spring 1997" respectively, while those in Figure 2(b) are "Introductory Microeconomics", "Economics 304K", and "Spring 2000" respectively.

As shown by these two course homepages, course ID & time usually conform to certain patterns, which can be expressed by *regular expressions* (the patterns for course ID & time, and their extracting process will be detailed later in Section 4.3). However, since online course pages are written in different formats by different authors and they cover a variety of disciplines, it is impossible to manually compose rules in terms of functions of dozens of features to extract course titles. Although a course title is usually shown prominently in terms of both position and format the degree of prominence for course titles can not be accurately defined by simple rules. For example, one might think that a course title is the field with the biggest font size in the course homepage. The example in Figure 2(b) actually invalidates this assumption. In this course homepage the field with the biggest font size is the course ID "Economics 304K", while the course title "Introductory Microeconomics" is below it in a smaller font. Therefore, in this work we adopt the statistic models proposed in this paper to extract course titles, and use pattern matching to extract course ID & time. The reasons why we do not incorporate the extraction of course ID & time into a unified graphical model will also be detailed in Section 4.3.

2.2 Problem Formulation

Usually, the input HTML Web page is represented in a DOM tree, and a leaf node in the DOM tree contains text and format information (in the following we call a leaf node in the DOM tree a DOM node for short). Here, we assume that a DOM node may correspond to only one kind of metadata. Although this assumption is true in most cases there are some exceptions as shown in Figure 2, which requires some preprocessing. The text "ECS - Data Structures - Spring 1997 (Sections A & B)" inside the red rectangle in Figure 2(a) is a DOM node, which contains all of the three kinds of course metadata: ID ("ECS 110"), title ("Data Structures") and time ("Spring 1997"). In this case, we have to split such a DOM node into multiple units. We conduct node splitting by turning all the candidates of course ID & time, which match the course ID & time patterns, into a new DOM node.

Therefore, a Web page can be represented as $x, x^1, \cdots, x^k$, where x is the input (observed) variable corresponding to the whole page and $x^1, \cdots, x^k$ are the k DOM nodes after preprocessing in this page. Based on the representation of a Web page, we can now formally define the concepts of Web Classification and Web Information Extraction as follows.

DEFINITION 1. (*Web Classification*) *Given a Web page, Web classification is the task of assigning a label to this page from a pre-defined set of labels.*

DEFINITION 2. (*Web Information Extraction*) *For all the DOM nodes after preprocessing, information extraction is the process of assigning metadata labels to these nodes.*

The problem of joint optimization for both Web classification and Web information extraction is defined as follows.

DEFINITION 3. (*Joint Optimization of Web classification and Web Information Extraction*) *Given a Web page represented as $\vec{x} = x, x^1, \cdots, x^k$, where x is the input variable corresponding to the whole page and $x^1, \cdots, x^k$ are the input variable of DOM nodes after preprocessing. Let $\vec{y} = y, y^1, \cdots, y^k$ be one possible label assignment for $x, x^1, \cdots, x^k$. By the principle of Maximum A Posteriori (MAP) the goal of Web classification and Web information extraction is to identify the label assignment y^* such that*

$$y^* = \arg\max_{\vec{y}} P(\vec{y}|\vec{x}). \quad (1)$$

Note that the label sets for classification and information extraction are different. Specifically, the label set for y contains all the categories of Web pages, while the label set for $y^1, \cdots, y^k$ contains all types of metadata. Based on the above definitions, the main difficulty is the calculation of the conditional probability $P(\vec{y}|\vec{x})$. In the next section we introduce a model of CRFs with constrained output to compute this probability.

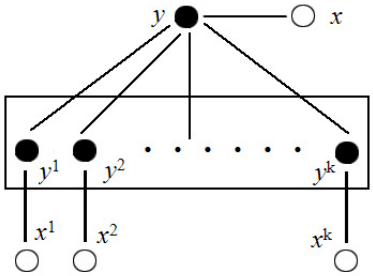


Figure 3: The graphical model for combing Web classification and Web information extraction

3. CRFS FOR TASK COMBINATION

In this section, we first introduce some basic concepts of CRFs and its simplest version for the traditional classification problems with a single output variable. Then, we describe our proposed model for integrating Web classification and Web information extraction and give the method to learn the parameters and perform prediction.

3.1 Preliminary

CRFs [5] is an undirected graphical model that is globally conditioned on observations. Let $g = (\vec{v}, \vec{e})$ be an undirected graph, where the node set can be partitioned into two groups $\vec{x}$ and $\vec{y}$. $\vec{x}$ are input variables over the observations, and $\vec{y}$ are output variables over the corresponding labels. The conditional distribution of the labels $\vec{y}$ given the observations $\vec{x}$ has the form,

$$P(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{x})} \prod_{c \in C} \varphi_c(\vec{x}_c, \vec{y}_c), \tag{2}$$

where C is the set of all the cliques in G, $\vec{y}_c$ and $\vec{x}_c$ are the components of $\vec{y}$ and $\vec{x}$ respectively, φ_c is a potential function with non-negative real values, and $Z(\vec{x})$ is the normalization factor with the form,

$$Z(\vec{x}) = \sum_{\vec{y}} \prod_{c \in C} \varphi_c(\vec{x}_c, \vec{y}_c). \tag{3}$$

The potential function can be expressed in terms of the feature functions $q_i(\vec{x}_c, \vec{y}_c)$ and their weights λ_i as

$$\varphi_c(\vec{x}_c, \vec{y}_c) = \exp(\sum_i \lambda_i q_i(\vec{x}_c, \vec{y}_c)). \tag{4}$$

Let $\vec{x} = x$ and $\vec{y} = y$ ($\vec{x}$ and $\vec{y}$ are both single-variable vectors), then $P(y|x)$ is simplified to

$$P(y|x) = \frac{\exp(\sum_i \alpha_i q_i(x, y))}{\sum_y \exp(\sum_i \alpha_i q_i(x, y))}. \tag{5}$$

When y is a discrete random variable, Equation 5, which is equivalent to Logistic Regression [3], can be used for classification.

3.2 CRFs for Two Joint Tasks

In the graphical model of Figure 3 (in this figure an open circle indicates that the variable is not generated by the model, and the edges among $y^1, \cdots, y^k$ are omitted), x and y are the observed and the output variables for a Web page respectively. The label space of y contains all the possible class labels for Web classification, and in our application, $y = \pm 1$, indicating whether it is a course homepage or not. $x^j (j = 1, \cdots, k)$ is the observed variable corresponding to a DOM node in the Web page x, and $y^j (j = 1, \cdots, k)$ is the output variable of x^j. The label space of y^j contains all the possible labels for information extraction, and in our application, $y^j = \pm 1$, indicating whether it is a course title or not. Then, the conditional probability can be expressed as

$$\begin{aligned} &P(\vec{y}|\vec{x}) = P(y, y^1, \cdots, y^k | x, x^1, \cdots, x^k) \\ =&\frac{\exp(E(y, y^1, \cdots, y^k, x, x^1, \cdots, x^k))}{\sum_{\vec{y}} \exp(E(y, y^1, \cdots, y^k, x, x^1, \cdots, x^k))}, \end{aligned} \tag{6}$$

where

$$\begin{aligned} &E(\vec{y}, \vec{x}) = E(y, y^1, \cdots, y^k, x, x^1, \cdots, x^k) \\ =&\sum_i \alpha_i f_i(x, y) + \sum_{j=1}^{k} \sum_r \beta_r g_r(x^j, y^j) + \sum_t \gamma_t h_t(y, y^1, \cdots, y^k) \\ =&\vec{\alpha}^T \vec{f}(x, y) + \sum_{j=1}^{k} \vec{\beta}^T \vec{g}(x^j, y^j) + \vec{\gamma}^T \vec{h}(y, y^1, \cdots, y^k), \end{aligned} \tag{7}$$

and f_i is the feature function applied to x and y for Web classification, g_i is the feature function applied to x^j and y^j on the j-th DOM node for Web information extraction ($j = 1, \cdots, k$), and h_i is the feature function on all the output variables $y, y^1, \cdots, y^k$. The h functions explicitly considers the dependency between y and the set of $y^1, \cdots, y^k$. This dependency is denoted by the lines between y and the set of $y^1, \cdots, y^k$ in Figure 3. This way, the h functions further bridge the labels of the Web page and all the DOM nodes, and Equations (6) and (7) become a general model for both Web classification and information extraction.

3.3 Model Inference with Constrained Output

Given a set of labeled data $\{(\vec{x}_l, \vec{y}_l)\}_{l=1}^n$, the model parameters $\vec{\alpha}$, $\vec{\beta}$ and $\vec{\gamma}$ can be estimated under the principle of *Maximum A-Posteriori*. Specifically, with the Gaussian prior the model learning problem is to find the models that maximize:

$$\mathcal{L}(\vec{\alpha}, \vec{\beta}, \vec{\gamma}) = \sum_{l=1}^{n} \ln P(\vec{y}_l|\vec{x}_l) - \frac{\lambda_1}{2}\|\vec{\alpha}\|^2 - \frac{\lambda_2}{2}\|\vec{\beta}\|^2 - \frac{\lambda_3}{2}\|\vec{\gamma}\|^2. \tag{8}$$

This concave function can be globally optimized by the nonlinear optimization methods, such as L-BFGS [6] (we adopt this method in this paper). The gradient vectors on $\vec{\alpha}$, $\vec{\beta}$ and $\vec{\gamma}$ are given as

$$\frac{\partial \mathcal{L}}{\partial \vec{\alpha}} = \sum_{l=1}^{n} (\vec{f}(x_l, y_l) - \frac{\sum_{\vec{y}} \vec{f}(x_l, y) \exp(E(\vec{y}, \vec{x}_l))}{\sum_{\vec{y}} \exp(E(\vec{y}, \vec{x}_l))}) - \lambda_1 \vec{\alpha}, \tag{9}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \vec{\beta}} =&\sum_{l=1}^{n} (\sum_{j=1}^{k_l} \vec{g}(x_l^j, y_l^j) - \frac{\sum_{\vec{y}} (\sum_{j=1}^{k_l} \vec{g}(x_l^j, y^j)) \exp(E(\vec{y}, \vec{x}_l))}{\sum_{\vec{y}} \exp(E(\vec{y}, \vec{x}_l))}) \\ &- \lambda_2 \vec{\beta}, \end{aligned} \tag{10}$$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \vec{\gamma}} =&\sum_{l=1}^{n} (\vec{h}(y_l, y_l^1, \cdots, y_l^{k_l}) - \frac{\sum_{\vec{y}} \vec{h}(y, y^1, \cdots, y^{k_l}) \exp(E(\vec{y}, \vec{x}_l))}{\sum_{\vec{y}} \exp(E(\vec{y}, \vec{x}_l))}) \\ &- \lambda_3 \vec{\gamma}, \end{aligned} \tag{11}$$

where E is defined in Equation (7), k_l in Equation (10) is the number of the DOM nodes in the l-th Web pages.

Given the model, we are interested in finding the most probable configuration of $\vec{y}$ in the label space for a given instance $\vec{x}$, as shown in Equation (1). This is the process of model inference.

The key problem in both model learning and inference is the computation of the normalization factor $\sum_{\vec{y}} \exp(E(\vec{y}, \vec{x}))$ since the label space of $\vec{y}$ is usually exponential to the length of $\vec{y}$. However, using the relation between the page type and the metadata we can significantly reduce the output label space. In our application of course homepage identification and course metadata extraction we assume that 1) a course homepage contains one and only one course title; 2) a non course homepage does not contain a course title.

These two assumptions are reasonable due to the following reasons. 1) Here, a course homepage is defined as the entry page of an online course, including all the information about the course. Thus, the course title is the indispensable element for a course homepage[2]. 2) A course homepage may contain the title string more than once, but we only consider the most prominent one in terms of format and place as the course title. 3) A non course homepage may talk about a course by mentioning the course name. Since the mentioned course name is not the key information in a non course homepage, it does not serve the purpose of a title by highlighting the format and position. So we do not consider this appearance of a course name as the metadata we want to extract. 4) A course listing page may list all the information of several different courses, including their titles, IDs and time. A course listing page is not considered a course homepage in this paper.

Thus, under these assumptions the label space of $y, y^1, \cdots, y^k$ has only $(k+1)$ elements:

$$(k+1)\begin{cases} y=-1, y^1=-1, y^2=-1\cdots, y^k=-1 \\ y=1, y^1=1, y^2=-1\cdots, y^k=-1 \\ y=1, y^1=-1, y^2=1\cdots, y^k=-1 \\ \cdots \\ y=1, y^1=-1, y^2=-1\cdots, y^k=1 \end{cases}$$

With this reduced label space the denominator $\sum_{\vec{y}} \exp(E(\vec{y}, \vec{x}))$ is efficiently tractable. Therefore, *exact inference* can be achieved in this model.

In our project we only define two h functions:

$$h_1(y, y^1, \cdots, y^k) = \begin{cases} 1 & \text{if } y=1, \text{ and } \exists j \text{ such that } y^j=1 \\ 0 & \text{otherwise.} \end{cases}$$

$$h_2(y, y^1, \cdots, y^k) = \begin{cases} 1 & \text{if } y=-1, \text{ and } y^j=-1 \text{ for } j=1,\cdots,k \\ 0 & \text{otherwise.} \end{cases}$$

These two h functions are actually defined to express the correlations between course homepage and course title.

3.4 Some Discussion

Different networked CRFs models, such as 2D CRFs [13], and hierarchical CRFs [14], have been proposed according to the different structures in $\vec{y}$ and $\vec{x}$. In some of these models the computation of the normalization factor of Equation 3 is often intractable, thus, *approximate inference* is often required in their inference process. However, a recent work by A. Kulesza and F. Pereira [4] shows that the learning of structured models can fail even with an approximate inference method with rigorous approximation guarantees. The proposed model of CRFs injects the constraints among output variables deep into it. This way, we greatly reduce the label space of $\vec{y}$, and thus enable *exact inference* to achieve optimal result.

The way to constrain the output label space is application specific, but we believe that this idea is generally applicable to a wide range of applications. For example, in another project, we are trying to classify Web pages as news, blog, product, recipe, and so on, and also extract metadata from the pages. The metadata for different type of pages are different. For example, we want to extract the title, author, and date from news pages but product name, picture, and price from product pages. If we use the model in this paper to conduct classification and extraction jointly, we can use the relation and constraints between the page type and metadata to significantly reduce the output label space to make the problem tractable.

4. FEATURES FOR THE JOINT MODEL

Since a Web page and a DOM node are different entities with different granularity we use two groups of features to characterize them separately.

4.1 Features for Course Homepage Detection

The course homepages usually contain some course terms, e.g., "*instructor*", "*teaching assistant*", "*course introduction*", "*course scheduling*", "*syllabus*" etc. These terms are the key features to distinguish course homepages from non course homepages. Additionally, we find that these terms often appear at a conspicuous place with some HTML formatting in a course homepage. To filter out noisy terms, the HTML format and position of a text node are considered. Specifically, we use the following heuristics to extract the candidates of course terms: 1) the course term candidate cannot be too long in terms of the number of English words in it; 2) the course term candidate cannot be located in the right block, top block and bottom block of a Web page; 3) the course term candidate is usually conspicuous in the web page in terms of its format (e.g., header decoration, bold and all capital etc.); 4) a term ending with a colon is a course term candidate.

Furthermore, we manually collected a collection of course terms and partitioned them into multiple groups. The course terms in the same group have similar semantics. Table 1 shows some groups of similar course terms. Each group of course terms corresponds to a feature for Web classification, and this feature is set to true when a course term candidate, extracted by the above heuristics from the Web page, appears in the corresponding term group. Some preliminary experiments show that the proposed features for Web classification is much better than treating the Web page as a bag of words.

4.2 Features for Course Title Extraction

To extract course title we use as many effective features as possible. All these features can be divided into three groups: *format features*, *position features* and *content features*. The format features, which are about font size, font color, decoration type and so on, are used to describe the characteristics that course title is usually the most prominent field

[2]There do exist some rare course homepages, which do not contain a course title on them. We ignore these course homepages in this work.

Table 1: The features for course homepage

No.	Course Terms
1	course name, course title, course id, course code, ...
2	instructor, co-instructor,course administrator, ...
3	teaching assistant, ta, teaching fellow, course assistant, ...
4	prerequisite, pre-requisite, course requirement, ...
5	course objective, course target, learning goal, ...
6	course outline, course overview, course summary, ...
7	course introduction, course description, ...
8	course schedule, course plan, course calendar, ...
9	course topic, course content, course document, ...
10	recommended reading, useful book, reference resource, ...
...	... ...

in the whole Web page or it is conspicuous locally with a different format from its neighboring texts. The position features are used to capture the heuristic that course title is often located in a noticeable position, such as the top half region of the Web page. The format features and position features are *application-independent*, however, the content features are *application-dependent*. Some typical characteristics addressed by the content features include: 1) course title cannot be too short or too long in terms of the number of English words in it; 2) course title cannot be the following expressions: Email, URL, Telephone, Time, combinations of letters and numbers; 3) Course title usually has a course ID or course time around it. Altogether we induce 34 features for course title extraction, including 29 application-independent features and 5 application-dependent features. The details of these features are omitted due to the space limitation.

4.3 Extraction of Course ID & Time

In general, a course ID is a combination of letters and numbers. Usually, its starting letter string is 1) the first letter of each keyword in the course title, 2) the name of the course discipline, 3) the abbreviation of the course discipline, 4) combination of 2) or 3) for multi-discipline courses. Following this starting letter string is the course code, comprising of numbers, letters, or delimiters. A course time usually contains information about year, semester, season or month. Therefore, Course ID & time can be extracted by matching these patterns. However, not all fields that match the specified patterns are course ID or course time (for example, other time information, such as publication time and exam time, may appear in the course homepage). That is, matching a pattern is a necessary condition but not sufficient to extract course ID & time. Nevertheless, we notice that course ID, course time and course title usually appear closely in a local area of the course homepage. Thus, we can leverage this heuristic to identify the true course ID & time among all the course ID & time candidates.

Based on the above analysis, we can extract course metadata as follows: 1) Find all the candidates of course ID & time, which match the corresponding patterns. 2) Extract course title by the proposed graphical model. 3) If Step 2 recognizes the given page as a course homepage and obtains the corresponding course title, identify the course ID & time candidates which are closest to the extracted course title, and output them as the extracted course ID & time.

Note that we do not incorporate the extraction of course ID & time into the graphical model. This is because: 1) As mentioned before, the candidates of course ID & time can be easily extracted by pattern matching. 2) Excluding course ID & time in the graphical model further simplify the label space of $y^1, \cdots, y^k$, thus help to achieve *exact inference* in model training and prediction. 3) We notice that there exist some correlations among course title, time and ID. Specifically, course ID, time and title usually appear closely in a local area of the course homepage. To take this point into account we introduce a feature for course title extraction to check whether a DOM node has a neighbor that is a course ID or time candidate. We believe that this method has the same effect as adding course ID & time to the label set of course metadata and defining the feature functions on the labels of neighboring DOM nodes. Therefore, this technique reduces the complexity of the graphical model without sacrificing performance.

This approach of using pattern-based method to extract some metadata and using machine learning to extract the rest is generally applicable in many applications. For example, if we want to build a vertical search engine for product search, the product metadata usually include product name, model number, price, and availability. Here, the model number, price, and availability can be characterized by patterns. Specifically, model number is a short sequence of characters, numbers, and special characters like "-"; price usually follows the $ sign and has two digits after the decimal point, and availability usually contains descriptions like "in stock", "out of stock", "available", "ships in 1-2 business days". On the other hand, the product name cannot be characterized by patterns and is better extracted by machine learning methods.

5. BASELINE METHODS

We compare our approach with two baseline methods. For each baseline method, we detail its training and prediction processes. For training, we are given a set $\mathcal{D}$ of labeled Web pages, including a set $\mathcal{D}_c$ of course homepages and a set $\mathcal{D}_n$ of non course homepages. For each course homepage in $\mathcal{D}_c$ its course title is also labeled. In these baseline methods the classification model in Equation (5), which is equivalent to Logistic Regression, is adopted.

5.1 Local Training and Separate Inference

In this method we train two classifiers, $\mathcal{C}_c$ and $\mathcal{C}_e$, for Web classification and Web information extraction respectively. $\mathcal{C}_c$ can use the features for Web classification (detailed in Section 4.1) to give the probability that a Web page is a course homepage, while $\mathcal{C}_e$ can use the features for Web information extraction (detailed in Section 4.2) to give the probability that a DOM node is a course title. Specifically, we use all the labeled Web pages in $\mathcal{D}$ to train $\mathcal{C}_c$, and we use the labeled DOM nodes from the course homepages in $\mathcal{D}_c$ to train $\mathcal{C}_e$.

In the prediction process we first use $\mathcal{C}_c$ to identify all the course homepages. Then, for each identified course homepage we use $\mathcal{C}_e$ to output the probability of each DOM node in this page that it is a course title. Finally, we select the DOM node with the largest probability in a course homepage as the node for the course title.

Since each course homepage contains only one course title (positive instance) while all of the rest are negative instances, the training data set for $\mathcal{C}_e$ is highly unbalanced. However, in Section 6.1 we will show that this property of unbalanced data does not affect the performance of course title extraction from the course homepages.

5.2 Local Training and Joint Inference

In this method we train the same two classifiers, $\mathcal{C}_c$ and $\mathcal{C}_e$, as those in Section 5.1. In the prediction process, given a Web page $x, x^1, \cdots, x^k$, we use a joint inference similar to the one in Section 3.3 to assign their labels. Specifically, we use the conditional probability in Equation (6), in which the function $E(\vec{y}, \vec{x})$ is replaced by

$$\begin{aligned} E(\vec{y}, \vec{x}) &= E(y, y^1, \cdots, y^k, x, x^1, \cdots, x^k) \\ &= \sum_i \alpha_i f_i(x, y) + \eta \sum_{j=1}^{k} \sum_r \beta_r g_r(x^j, y^j) \\ &= \vec{\alpha}^T \vec{f}(x, y) + \eta \sum_{j=1}^{k} \vec{\beta}^T \vec{g}(x^j, y^j), \end{aligned} \tag{12}$$

where $\vec{\alpha}$ and $\vec{\beta}$ are actually the parameters in $\mathcal{C}_c$ and $\mathcal{C}_e$, and $\eta > 0$ is the weight to balance the tradeoffs between the two models. The larger the value of η is, the more influence the information extraction model $\mathcal{C}_e$ has on the final result. The smaller the value of η is, the more influence the classification model $\mathcal{C}_c$ has on the final result. In this formulation we ignore the h functions in Equation (7).

In summary, the method in Section 5.1 performs local training and separate inference, while the method in Section 5.2 conducts local training and joint inference. Compared with these baseline methods our approach conducts joint learning in the training process and joint inference in the prediction process.

6. EXPERIMENTAL EVALUATION

In this section, we report empirical results by applying our joint model to perform classification and information extraction simultaneously. We compare our approach with the two baseline methods, detailed in Section 5. For the two classifiers $\mathcal{C}_c$ and $\mathcal{C}_e$ used in the two baseline methods, we carefully tune their parameters used in the training process and compare their best performances with those of the proposed model. The results show that our model significantly improves both Web classification and Web information extraction.

To evaluate all these methods, we collected the labeled data set $\mathcal{D}$, including the set $\mathcal{D}_c$ of 530 course homepages and the set $\mathcal{D}_n$ of 1200 non course pages, and manually annotated the course title, ID and time for each course homepage.

For both Web classification and Web information extraction, we use the standard *accuracy*, *precision*, *recall* and *F1* measures to evaluate these methods. For course homepage identification these metrics are calculated using all the data in our data set $\mathcal{D}$. For course title extraction the evaluation data sets are different, depending on the extraction method and the situation. Specifically, when evaluating the performance of $\mathcal{C}_e$ in course title extraction, we assume that the inputs of $\mathcal{C}_e$ are all course homepages. Thus, the corresponding evaluation metrics are calculated within all the course homepages. However, when evaluating the joint model for course title extraction, the inputs to this model include both course homepages and non course pages, so the evaluation is conducted with both types of pages. Since our goal is to find all the course titles accurately from a mixture of course homepages and non course pages, the metrics calculated within this mixed data set are the most important measures we care about. Since we aim to show the effectiveness of the proposed graphical model we omit the experimental results of course ID & time extraction in this section. Note also that all the experimental results in this section are obtained from 10-fold cross validation.

Table 2: The experimental results of $\mathcal{C}_c$ for course homepage identification

λ	Accuracy	Precision	Recall	F1
0	0.949	0.903	0.821	0.858
0.1	0.949	0.902	0.818	0.856
1	0.95	0.919	0.807	0.857
5	0.947	0.929	0.775	0.844
10	0.947	0.936	0.768	0.842
20	0.943	0.952	0.736	0.828
100	0.924	0.980	0.607	0.747

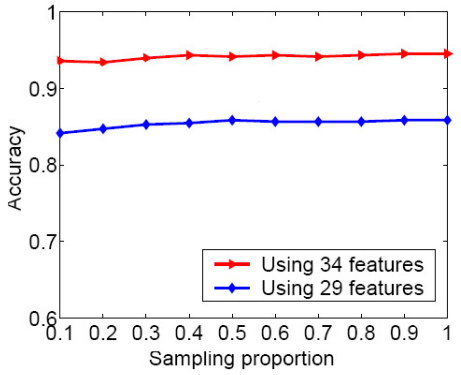


Figure 4: The accuracy of $\mathcal{C}_e$ for course title extraction from course homepages

6.1 Results on $\mathcal{C}_c$ and $\mathcal{C}_e$

The evaluation results of the classifier $\mathcal{C}_c$ for course homepage identification are listed in Table 2, where λ is the parameter on the penalty term to compensate for the overfitting of complex models. These results show that 1) when $\lambda = 0$ it achieves the best result of F1 value; 2) Along with the increase of λ values, the precision increases while the recall decreases. We have conducted more theoretical analysis on the relationship between λ and the performances of precision and recall, however, it is out of the scope of this paper.

We evaluate the performance of $\mathcal{C}_e$ for course title extraction under the assumption that $\mathcal{C}_e$ performs only on course homepages. Thus, we use only the 530 course main pages to conduct the 10-fold cross validation. Note again that the DOM tree for each page contains only one node that is the course title (the positive instance) but many more (several hundred) negative instances, so the data set is very unbalanced. To alleviate this problem, we under-sample the negative instances by a certain ratio p in each round of cross validation, but keep all the positive instances. The experiments in this part aim to answer the following questions: 1) What is the relationship between the sampling proportion p and the extraction accuracy? 2) How much does the extraction accuracy of course title improve with the 5 application-dependent features?

To answer these questions, we measure the performance of $\mathcal{C}_e$ with two groups of features (all of the 34 features and

Table 3: The experimental results on the two baseline methods

Baseline Methods	Course Homepage Identification				Course Title Extraction			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Local training and Separate Inference	0.949	0.903	0.821	0.858	0.941	0.855	0.774	0.811
Local training and Joint Inference ($\eta = 1$)	0.934	0.801	0.871	0.833	0.927	0.764	0.832	0.795

Table 4: The experimental results on the joint optimization method

$(\lambda_1, \lambda_2, \lambda_3)$	Course Homepage Identification				Course Title Extraction			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
(0,0,0)	0.963	0.922	0.879	0.898	0.953	0.863	0.825	0.842
(1,1,0.001)	0.967	0.954	0.864	0.906	0.956	0.889	0.807	0.846
(5,5,5)	0.965	0.968	0.839	0.898	0.955	0.904	0.786	0.840
(10,10,0.1)	0.963	0.972	0.829	0.894	0.952	0.900	0.768	0.828
(10,10,10)	0.963	0.972	0.829	0.897	0.952	0.900	0.768	0.828

the 29 application-independent features only) and ten different under-sampling ratios: $0.1, 0.2, \cdots, 1$. The results are shown in Figure 4. we give the experimental findings and the corresponding analysis as follows: 1) The application-dependent features significantly improve the extraction accuracy of course title. This improvement is independent of the value of the under-sampling ratio. On average, the performance of $\mathcal{C}_e$ increases 8.72% by adding the application-dependent features. 2) $\mathcal{C}_e$ yields almost the same performance under different under-sampling ratio. The reason for this is that different degrees of data imbalance may change the absolute value of probability that a node is a course title, however, it will not change the ranking of the nodes in the decrease order of this probability. Therefore, the node with the biggest probability value in a course Web page remains the same under different imbalance settings.

Note that in the above experiments the parameter λ used in training $\mathcal{C}_e$ is set to 0, and we found that different values of this parameter give similar results.

6.2 Results of Web Classification and Web Information Extraction on Any Web Pages

In this subsection we evaluate the performance of the joint optimization model and the two baseline methods for course homepage identification and course title extraction given a mixture of course homepages and non course pages. Thus, we use all the Web pages in $\mathcal{D}$ to conduct the 10-fold cross validation.

As shown in Sections 6.1, $\mathcal{C}_c$ and $\mathcal{C}_e$ achieve their best performances respectively when the parameters λ are both set to 0 in their training. Thus, we adopt these parameter settings in training $\mathcal{C}_c$ and $\mathcal{C}_e$ for the two baseline methods. Furthermore, for the second baseline method we evaluate its performance under different values of the tradeoff parameter η in Equation 12. The results show that setting η to 1 achieves the best F1 value for both course homepage identification and course title extraction. Thus, we only report this best performance for the second baseline method.

The performances of these two baseline methods are shown in Table 3. It shows that 1) the first method is slightly better than the second one in term of the F1 value for course title extraction; 2) the second method increases the recall for course title extraction at the cost of precision.

The performances of our joint optimization method with different settings of $\lambda_1, \lambda_2, \lambda_3$ in Equation 8 are recorded in Table 4. We also conduct the t-test on the results from the 10-fold cross validation to check whether the proposed model significantly outperforms the baseline methods with a 95% confidence interval. The results show that 1) under all these parameter settings the proposed method is significantly better than the two baseline methods in terms of the F1 values for both course homepage identification and course title extraction; 2) under all the parameter settings this joint model significantly outperforms the first baseline method in terms of all the evaluation metrics, including precision, recall and F1 value; 3) the parameter setting (1,1,0.001) achieves the best performance of course title extraction in terms of F1 value. In this parameter setting we intentionally set λ_3 (the parameter on the h functions) to a very small value in order to increase the effect of the h functions in this model. It shows that this setting slightly improves the performance of this model.

7. OFCOURSE PORTAL

We have applied this combination model to the *OfCourse* portal for online course search. This portal currently contains over 60, 000 courses from the top 50 schools in the US. The current portal supports basic keyword search, advanced search, and browsing by subject areas. In the advanced search, the search can be conducted within the scope of the user-specified university and year. For each course in the search result, the extracted course title is displayed, together with the school, the extracted year, and an excerpt from the course page containing the search term. The course title links to the course homepage on the Web. These functions are all supported by the combination model proposed in this paper. Some user studies show that these results of classification and information extraction are quite satisfactory. Additionally, the totally automatic process of course homepages identification and course metadata extraction allows us to easily make our portal an open system, in which any user can add new courses into the portal by only submitting the URLs. The model behind can identify if the submitted URL is a course homepage and simultaneously extract its course metadata if it is. Interested users can experience the effectiveness of our joint model by submitting any URL at the *OfCourse*[3] portal.

8. RELATED WORK

The works most related to course title extraction are title extraction from Web pages [12]. The assumption used in [12] is that a general title is usually the most conspicuous field in any Web page. However, the likelihood that this assumption is true is much lower for the course titles of online courses. To achieve high precision in course metadata extraction we propose some application-dependent features, which implicitly consider the relationships among the features and labels of the neighboring DOM nodes.

[3]http://fusion.hpl.hp.com/OfCourse/addNewCourse.jsp

All of these previous works perform under the assumption that we have a perfect classifier, which can precisely and completely identify the target Web pages (always containing the metadata we want to extract) from a mixture of Web pages. However, this assumption is not always true. The motivation examples in Section 1.1 intuitively show the room of improvement in Web classification by considering the backward dependency from Web information extraction to Web classification. Thus, we seamlessly combine these two tasks into one graphical model to mutually boost their performances.

The methodology to combine multiple mutually-dependent tasks into one statistical graphical model appears in some previous works for different applications. Roth et al. focus on the topic of learning and inference over structured and constrained output with the applications of natural language processing [11, 9]. The work in [11] combines the tasks of entity and relation recognition, in which separate classifiers are first trained for entities and relations, and then these local classifiers are used to make global inferences for the most probable assignment for all entities and relations of interest. It actually motivates the baseline method in Section 5.2. Another related work is the integration of data record detection and attribute labeling for online product extracting from product Web pages by Hierarchical CRFs [14]. Both the learning and inference in this work are jointly optimized. The work in this paper further emphasizes the difference between local and joint training, and the difference between local and joint inference. A recent paper [1] from Bhattacharya et al. deals with the combination of structured entity identification and document categorization. The combined two tasks in that paper are different from those in this paper. Additionally, they use the generative model in training and prediction, however, we use the discriminative model CRFs to achieve the combination.

9. CONCLUSIONS

This paper discussed the problem of combining Web classification and Web information extraction, which are actually two coupled steps for Web content analysis. Previous works in this area treat them as two separate and sequential steps, which consider the forward dependency from Web classification to Web information extraction, but ignore the backward dependency from Web information extraction to Web classification. To mutually boost these two steps, we propose to combine them by using a model of CRFs. Specifically, this graphical model contains two kinds of observed random variables with different granularity: the variable of the whole Web page for Web classification, and the variables of the DOM nodes inside the Web page for Web information extraction. This way these two steps are formulated as a joint task about assigning labels to these variables. Additionally, the constraints among the two kinds of random variables can be utilized to simplify the learning and inference process of this model. The experiments in our project *OfCourse* for online course search show that our joint model significantly outperforms the two baseline methods in terms of F1 value. Even though this approach is designed to solve the specific problem of course homepage identification and course metadata extraction, the methodology of combining Web classification and Web information extraction is problem-independent and can be applied to many other applications.

10. ACKNOWLEDGMENTS

The authors thank Baoyao Zhou for providing part of the test data and a Web page segmentation tool. We are also grateful to Shicong Feng and Liwei Zheng for helpful collaborations in the *OfCourse* project. The authors Fen Lin and Zhongzhi Shi are partially supported by the National Science Foundation of China (60775035), 863 National High-Tech Program (No.2007AA01Z132), and National Basic Research Priorities Programme (No. 2003CB317004, 2007CB311004).

11. REFERENCES

[1] I. Bhattacharya, S. Godbole, and S. Joshi. Structured entity identification and document categorization: two tasks with one joint model. In *Proc. of the 14th ACM SIGKDD*, 2008.

[2] M. Castellanos, Q. Chen, U. Dayal, M. Hsu, M. Lemon, P. Siegel, and J. Stinger. Component adviser: a tool for automatically extracting electronic component data from web datasheets. In *Proc. of the Workshop on Reuse of Web-based Information, the 7th WWW*, 1998.

[3] D. Hosmer and S. Lemeshow. *Applied Logistic Regression*. Wiley, New York, 2000.

[4] A. Kulesza and F. Pereira. Structured learning with approximate inference. In *Proc. of the 21st NIPS*, 2007.

[5] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. of the 18th ICML*, 2001.

[6] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[7] A. McCallum. Information extraction: Distilling structured data from unstructured text. *ACM Queue*, 2005.

[8] Z. Nie, J. Wen, and W. Ma. Object-level vertical search. In *Proc. of the Conf. on Innovative Data Systems Research*, 2007.

[9] V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Learning and inference over constrained output. In *Proc. of the 19th IJCAI*, 2005.

[10] J. Rennie and A. McCallum. Using reinforcement learning to spider the web efficiently. In *Proc. of the 16th ICML*, 1999.

[11] D. Roth and W. Yih. Probabilistic reasoning for entity and relation recognition. In *Proc. the 19th COLING*, 2002.

[12] Y. Xue, Y. Hu, G. Xin, R. Song, S. Shi, Y. Cao, C.-Y. Lin, and H. Li. Web page title extraction and its application. *Information Processing and Management*, 43(5):1332–1347, 2007.

[13] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. 2d conditional random fields for web information extraction. In *Proc. of the 22nd ICML*, 2005.

[14] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous record detection and attribute labeling in web data extraction. In *Proc. of the 12th ACM SIGKDD*, 2006.