Minimizing the Costs of the Training Data for Learning Web Wrappers

Rolando Creo, Valter Crescenzi, Disheng Qiu, Paolo Merialdo

Dipartimento di Informatica ed Automazione
Università degli Studi Roma Tre
Via della Vasca Navale, 79 – Rome, Italy

{creor, disheng, crescenz, merialdo}@dia.uniroma3.it

ABSTRACT

Data extraction from the Web represents an important issue. Several approaches have been developed to bring the wrapper generation process at the web scale. Although they rely on different techniques and formalisms, they all learn a wrapper given a set of sample pages. Unsupervised approaches require just a set of sample pages, supervised ones also need training data. Unfortunately, the accuracy obtained by unsupervised techniques is not sufficient for many applications. On the other hand, obtaining training data is not cheap at the web scale. This paper addresses the issue of minimizing the costs of collecting training data for learning web wrappers. We show that two interleaved problems affect this issue: the choice of the sample pages, and the expressiveness of the wrapper language. We propose a solution that leverages contributions in the field of learning theory, and we discuss the promising results of an experimental evaluation of our approach.

1. INTRODUCTION

The huge amount of information available on the web inspired several researches towards the development of tools and techniques to infer web wrappers for extracting data from script-generated HTML pages.

Unsupervised approaches take as input a set of sample pages and analyze regularities and differences to infer a wrapper based on the underlying HTML template [6, 2]. They could scale on the number of sources, but the accuracy of the generated wrappers is limited. Supervised approaches can produce accurate wrappers, but their scalability is limited because they need training data, i.e. annotations over the values published in the pages. In early approaches these data were provided by means of a human intervention [9]. More recently, solutions that rely on data stored in existing repositories have been proposed [7]. Unfortunately, in many domains suitable training data does not exist at all (consider pages that publish subjective values, such as customer ratings, or real time data, such as stock quote prices), or they might be biased over specific instances (typically the most popular). This prevents the generation of a correct wrapper around the broader set of instances, and thus raises the need of additional training data.

Overall, obtaining training data represents a relevant cost in the wrapper inference process. This paper presents a framework to minimize the the cost of generating a wrapper expressed as the number of *membership queries* (*MQ*) needed by a supervised inference system for the training [1]. Membership queries are the simplest form of queries since they admit only a yes/no answer, e.g. "*is* `rick@usr.edu` *a value to extract?*". It is worth observing that the simplicity of these queries make them suitable to be answered through crowdsourcing platforms, that make the training costs explicit.

As the following example illustrates, these costs depend on two interrelated features: the representativeness of the sample and the expressiveness of the extraction language.

The Sampling Problem. Suppose we are interested to wrap pages containing information about professors. For the sake of simplicity, let us represent pages as tables, where data is organized in rows and columns. Figure 1(a) shows sample pages depicted according to the above simplification. A wrapper can be described as a set of extraction rules. In our abstraction, an extraction rule specifies the cell containing the relevant data, and it can be expressed by *absolute* coordinates (e.g. first row, second column), or by *relative* coordinates, that is, with respect to another cell (e.g. the first cell located at the right of the cell containing 'Email'). Correspondingly, suppose that we choose to adopt only XPath extraction rules of one form out of two possible: *absolute* extraction rules (e.g. /html[1]/table[1]/tr[*x*]/td[*y*]) that we denote abs(*x*,y); *relative* extraction rules (e.g. //tr[td[contains(.,'*x*')]]/td[2]) that we denote right-of('*x*').

For example, according to *Rick*'s page, candidate extraction rules for Name are abs(1,1) and above('Home'). Similarly, rules for Position are abs(5,2) and right-of('Position').

Now suppose that the Position of professors is a relevant attribute to extract. If the sample set is composed only of awarded professors (such as *Rick*), inferred rules could not work for the broader set of all professors, including those without any award (such as *Mark* and *Bill*). For example, the rule abs(5,2) for Position might work for all the awarded

VLDS'12 August 31, 2012. Istanbul, Turkey.

Rick	
Home	rick.usr.edu
Awards	Best paper
Affiliation	CS Dept. | Univ. of R.
Position	Full Prof.
Email	rick@usr.edu
Phone	123-345-6789

Mark		
Home	mark.usm.edu	
Affiliation	IT Dept.	Univ. of M.
Position	Ass. Prof.	
Email	mark@usm.edu	
Phone	987-654-3210	

Bill		
Affiliation	SE Dept.	Univ. of G.
Position	Ass. Prof.	
Email	bill@usg.edu	
Phone	123-454-3210	

v_0^+ = mark@usm.edu

(a) sample pages

r_1	abs(5,2)
r_2	right-of('Email')
r_3	above('987-654-3210')
r_4	below('Ass. Prof.')

(b) extraction rules

	r_1	r_2	r_3	r_4
p_r	Full Prof.	rick@usr.edu	*nil*	*nil*
p_m	mark@usm.edu	mark@usm.edu	mark@usm.edu	mark@usm.edu
p_b	123-454-3210	bill@usg.edu	*nil*	bill@usg.edu

(c) values extracted

Figure 1: Running Example

professors, but it does not extract the position for other professors.

The usual approach to address this issue is to work with a large set of annotations that hopefully covers all the possible types of target pages. However, according to our cost model, this strategy is inefficient.

The Expressiveness Problem. Consider again the running example and suppose we are interested to extract professors' Name and Home. We now show that the size of the sample set actually depends on the expressiveness of the language used to specify the rules, and hence on the set of available rules.

Suppose that we choose to adopt only *absolute* extraction rules: a correct rule for Name is abs(1,1). Note that only one labeled sample would suffice to infer this rule. Suppose now to adopt a more expressive language, which also includes *relative* rules. Using just one page, say *Rick*'s, several rules are generated to extract Name: abs(1,1), above('Home'), above('rick.usr.edu'). To determine the correct rule at least another well chosen example is required: only with the help of *Mark*'s page we have the evidence that the rule above('Home') does not work.

To summarize, the more expressive is the model, the larger is the size of the representative sample set [1] [5] (intuitively, the space of hypotheses is larger and thus more examples are needed to discard the incorrect ones). However, the additional expressiveness should be carefully handled, since its actual need depends on the input pages and on the desired attributes, whereas it always entails additional costs.

The usual approach to address this issue is to work with overly expressive languages to cover all needs. However, as made explicit by our cost model, this strategy is inefficient.

Overview. In this paper we address the above issues. We show that they cannot be tackled separately, and propose an approach that carefully handles the expressiveness of the wrapping language, and the choice of the samples (i.e. the pages to be annotated).

We propose an approach, inspired by a statistical learning technique [14, 13], in which the expressiveness of the language is enlarged at runtime. We organize the class of candidate rules $\mathcal{R}$ into a hierarchy of classes $\{\mathcal{R}^h\}_{0 \leq h \leq m}$ of increasing size: initially the correct rule is searched only within $\mathcal{R}^0$, and then the set of available extraction rules might be expanded to a larger class $\mathcal{R}^h$.

The goal is to achieve the expressiveness of the largest classes only whenever it is detected as actually needed. For all those cases that can be solved within the smaller classes of rules, the algorithm will save many samples.

The approach is independent of the details of the formalism used to express extraction rules. Our hierarchy $\{\mathcal{R}^h\}$ makes use of simple rules, as follows: $\mathcal{R}^0$ is the *class of absolute XPath rules* as described before; $\mathcal{R}^h$, for $0 < h \leq m$, is obtained by adding to $\mathcal{R}^{h-1}$ the *class of relative XPath rules* with distance at most h from a pivoting leaf to the value.[1] However this is just one possibility: the hierarchy can be built with other of rules.

A landmark decision is whether and when expanding the set of candidate rules. We introduce a probabilistic model to dynamically characterize the probability of correctness for rules in the *current* class of candidate rule $\mathcal{R}^h$. We propose an original active learning algorithm [12] that exploits the probabilistic model for deciding to enlarge $\mathcal{R}^h$ *lazily*, i.e. only whenever there is enough evidence that the correct rule is not amongst the current set of candidates. The algorithm *actively* chooses the next membership query to find a representative sample set: it selects a sample page and poses a membership query on an extracted value. By accurately choosing this value, the number of queries can be minimized.

The paper is organized as follows: Section 2 discusses related work; Section 3 formalizes our setting and states the problem definition; Section 4 develops a probamilistic model to characterize the correctness of extraction rules; Section 5 presents an active learning algorithm for extraction rules based on the model; finally, Section 6 discusses our preliminary experiments with a set of sources from the Web.

2. RELATED WORK

In machine learning, the number of labeled samples needed by a supervised learning algorithm to infer a *good* hypothesis is called *sample complexity*, and has been studied from several perspectives. For instance, similarly to our setting, [1] discusses the problem of *exactly* inferring a *concept*, i.e. a set of elements, by means of *membership queries*, i.e. question

[1]In the current prototype all textual leaves are used as candidate pivot. The distance from the pivot to the extracted node is measured according to the number of edges crossed in the DOM representation of the HTML pages but considering contiguous siblings at distance 1.

of the type "*is this an element of the target concept?*". However, the main idea underlying our approach has been proposed by the statistical learning community [14], in which a *loss* function is given in order to characterize the quality of the produced hypothesis. The *structural risk minimization* (SRM) technique [13], i.e. the decomposition of the set of hypotheses into a hierarchy of subclasses, aims at avoiding the *overfitting* problem: since the class of hypotheses studied by this community might be so expressive to be able to arbitrarily reduce the loss, a trade-off with other quality criteria is needed to avoid that the learning algorithm selects the hypothesis describing the training data perfectly, rather than their underlying patterns.

Many researchers have proposed several variations of the learning paradigm to make it practically feasible in different applicative contexts: the learning approaches in which the inference algorithm is free to choose which sample to label next are usually defined *active* [12]. These have recently gained interest, since, as clarified in [3], that they might produce exponential improvements over the number of samples wrt traditional supervised approaches.

To the best of our knowledge and differently from our proposal, all the approaches for inferring wrappers over structured websites developed by the researchers in the *wrapper* community [6, 2, 9, 4, 15], define the set of hypotheses *statically*, i.e. before performing the inference, and once set, the set of candidate rules cannot be changed without seriously revisiting the inference algorithm. Therefore they oversize the expressiveness of the formal language used to specify the extraction rules and additional samples are required only to compensate with the excess of expressiveness.

Active learning approaches for wrapper induction have been proposed in [11, 10]. However, also in these works the expressiveness is *statically* defined. Moreover, the latter approach requires complex user interaction, since the user has to choose the correct wrapper within a set of ranked proposals.

A few recent proposals try to scale the wrapper inference to the web scale [7, 8]. In [7] the authors leverage an available dataset, but it is not clear how they can ignore the presence of biased samples (as suggested by its running example based on popular objects itself), while in [8] it is needed domain knowledge that only an human expert can provide.

3. PROBLEM DEFINITION

Preliminary Definitions: Let $U = \{p_1, p_2 \dots p_n\}$ be a set of pages. Every page publishes several attributes of interest (e.g. professor Position, Email, etc.). For simplicity, we develop the discussion concentrating on one attribute, and we assume that its values are either a textual leaf of the DOM tree representation of the pages, or a distinguished *nil* value. We write $v \in p$ to denote that v is a value of the page p, p_v to denote the page in which the value v is located, and $|p|$ to denote the number of values in p.

We refer to a generic *extraction rule* (or simply *rule*) r over the set of pages U as a concrete tool to build a vector of values indexed by the pages in U such that $r(p) \in p \cup \{nil\}$. Every rule extracts one vector of values from U denoted $r(U)$. Figure 1(c) shows the vectors extracted by the rules r_1, r_2, r_3, r_4 in Figure 1(b). We denote with $R(U)$ the set of vectors obtained by applying a set of rules R over U, and blur the distinction between a rule and the vector it extracts from U. Note that $|R(U)| \leq |R|$, with the strict inequality holding whenever a vector is extracted by different rules.

We denote with $\{\mathcal{R}^h\}$ the hierarchy of classes of all the *generable* rules, i.e. the extraction rules that can be generated to extract vectors from U. The classes of this family are countable but potentially not finite. We will manage to work with a finite restriction of them, called the set of *generated* rules, as discussed in the following.

Labeled Sample Sequences: we introduce the concept of *labeled sample* v^l where $v \in p_v$ is a value from a page p_v, and $l \in \{+, -\}$ is either a positive or a negative label. In the following v^+ and v^- denote a positive sample (or annotation) or a negative sample, respectively, that is the two possible answers to a *MQ*.

A rule r is *admissible* wrt to a set of samples L (denoted $L(r)$) iff:

$$L(r) \;\Leftrightarrow\; \forall v^l \in L,\; \begin{array}{l} l = + \rightarrow r(p_v) = v \\ l = - \rightarrow r(p_v) \neq v \end{array}$$

that is, it is compliant with the labels in the set.

The concept can be trivially extended to set of rules R, and we denote with $R_L = \{r \in R : L(r)\}$ the subset of admissible rules in R wrt L and with $\widehat{V}^R_L(U)$ all the values they extract from U: $\widehat{V}^R_L(U) = \{v : v = r(p), r \in R_L, p \in U\}$.

Example: *Let p_r, p_m and p_b be the pages in Figure 1(a) and let $U = \{p_r, p_m, p_b\}$. The attribute* Email *is extracted by the rule r_2 =*right-of('Email')*: two positive samples are v_0^+ =*'mark@usm.edu' *and v_1^+ =*'bill@usg.edu'*, a negative sample is v_2^- =*'123-454-3210'*. Observe that r_2 is admissible wrt to $L = \{v_0, v_1, v_2\}$. Now consider another rule r_1 =*abs(5,2) *and the set of rules $R = \{r_1, r_2\}$. Then r_1 is not admissible wrt to L since $r_1(p_b) = '123 - 454 - 3210'$ which is the negative sample v_2^-. Hence, $R_L = \{r_2\}$ and $\widehat{V}^R_L(U) = \{v_0, v_1\}$.*

In the following, given a set of rules R, we will only consider special ordered sets of samples, called *labeled sample sequences*, which are formed by an initial annotation, and then by adding only new values which are still admissible with respect to the samples already seen. Intuitively, a labeled sample sequence is the list of answers to the *MQ* posed to learn a rule:

Definition: *A* Labeled Sample Sequence *(l.s.s.) L wrt to a set of rules R and a set of pages U is specified by a sequence of labeled sample $v_0, \dots, v_k, \dots$ that defines a sequence of (observed) sets L^k with $L^{k+1} = L^k \cup \{v_k\} = \{v_0^+, v_1, \dots, v_k\}$ such that: (i) it begins with an annotation $v_0^+ \neq nil$, and (ii) $\forall k > 0$, $v_k \in V^R_{L^k}(U) = \widehat{V}^R_{L^k}(U) \setminus L^k$.*

The constraint (i) on the first annotation v_0^+ of the sequence is useful to get finite[2] R_{L^1}, whereas the constraint (ii) on the remaining samples entails that the *new* sample v_k that forms L^{k+1} from L^k leads to smaller and smaller admissible sets: $R_{L^{k+1}} \subseteq R_{L^k}$. It is worth noting that $R_{L^{k+1}}$ plays the role of what the learning communities call the *version-space* [12], i.e. the set of hypotheses still plausible after having considered an input set of labeled samples.

In the following we will uniformly refer to both L and one of its observed subsets L^k blurring the differences between the two concepts whenever the context clarifies which one is actually involved.

[2]The first annotation can also be conveniently seen as the specification of the desired attribute.

It can always be decided whether a rule extracting the desired vector exists. However, since it is not known in advance whether that rule was in the set of all candidate rules, the only certain way to be sure of its presence is by checking every single page [1]. In order to minimize the number of *MQ*, we develop a probabilistic characterization of the rules, and restate our problem in that sense.

Problem Definition: *Given a hierarchy of classes of extraction rules* $\{\mathcal{R}^h\}_{0\leq h\leq m}$ *over a set of pages* U*, and a threshold* δ*, find either a correct rule* $r \in \mathcal{R}^m$*, or conclude that no rule is correct in any* $\mathcal{R}^h$ *with probability greater than* $1-\delta$*, by minimizing the length of the input l.s.s.* L*.*

4. A PROBABILISTIC MODEL FOR STRUCTURAL RISK MINIMIZATION

We introduce a probabilistic model for evaluating the correctness of an extraction rule, given a l.s.s. L, and a class of rules $\mathcal{R}$. As a consequence, the model is able to compute the probability that a correct extraction rule has not been generated.

The notations used for main events covered by our analysis, and their probabilities, are summarized in Table 1. 5 We assume that the probability of an extraction rule is determined by the extracted values $r_1(U) = r_2(U) \Rightarrow P(r_1) = P(r_2)$.

Given a new labeled sample v_k^l to form $L^{k+1} = \{v_k^l\} \cup L^k$, we denote with $P(L^{k+1})$ the probability $P(v_k^l, L^k)$. By applying Bayes' theorem, the probabilities of the two main events of interest are:

$$P(r|L^{k+1}) = \frac{P(v_k^l|r, L^k)P(r|L^k)}{P(v_k^l|L^k)} \quad (1)$$

$$P(\overline{\mathcal{R}}|L^{k+1}) = \frac{P(v_k^l|\overline{\mathcal{R}}, L^k)P(\overline{\mathcal{R}}|L^k)}{P(v_k^l|L^k)} \quad (2)$$

where $P(v_k^l|L^k)$ is a *normalization factor* that can be expressed as:

$$\sum_{r\in\mathcal{R}_{L^k}} P(v_k^l|r_i, L^k)P(r_i|L^k) + P(v_k^l|\overline{\mathcal{R}}, L^k)P(\overline{\mathcal{R}}|L^k)$$

For any k, $P(r|L^{k+1})$ and $P(\overline{\mathcal{R}}|L^{k+1})$ can be defined iteratively by means of $P(v_k^l|r, L^k)$, $P(v_k^l|\overline{\mathcal{R}}, L^k)$, $P(\overline{\mathcal{R}}|L^k)$ and $P(r|L^k)$. $P(v_k^l|r, L^k)$ and $P(v_k^l|\overline{\mathcal{R}}, L^k)$ can be defined by abstracting the actual process that leads to the observed l.s.s into a simple *generative model*. This is essentially equivalent to define a p.d.f. over every l.s.s..

By repeatedly applying the bayesian updating rules expressed by equations 1 and 2, the model allows the computation of $P(\overline{\mathcal{R}}|L^{k+1})$ and $P(r|L^{k+1})$ for any k, starting from *prior*-probabilities $P(\mathcal{R}|L^0) = \mathcal{P}(\mathcal{R})$ of having generated a correct rule, and $P(r|L^0) = \mathcal{P}(r)$ of r being a correct rule. The iteration continues until admissible rules exist, i.e. until $\mathcal{R}_{L^k} \neq \emptyset$.

4.1 Generative Model

In this section we describe a simple generative model for the l.s.s., useful for deriving the posterior probability $P(r|L^k)$ that $r(U)$ is the correct vector to extract once a l.s.s. L^{k+1} has been observed. This vector is not known in advance, but the values forming the l.s.s. L^{k+1} will be labeled as either positive or negative according to it.

Let $\mathcal{P}(\mathcal{R})$ be the prior probability that the correct vector can be extracted by a rule belonging to $\mathcal{R}$. We suppose that the acquisition of a new labeled sample v_k to form L^{k+1} from L^k follows a uniform p.d.f. amongst all values still queryable, i.e. the values in $V_{L^k}^{\mathcal{R}}(U) = \widehat{V}_{L^k}^{\mathcal{R}}(U) \setminus L^k$.

Similarly, given a correct rule r, let $V^+(L^k, r) = V_{L^k}^{\mathcal{R}}(U), r(U)$ denote the set of all and only the values that can form new positive samples, and $V^-(L^k, r) = V_{L^k}^{\mathcal{R}}(U) \setminus V^+(L^k, r)$ the set of values that can form negative samples. It follows:

$$P(v_k^l|r, L^k) = \begin{cases} \frac{1}{|V_{L^k}^{\mathcal{R}}(U)|} & \text{, iff } v_k \in V^l(L^k, r) \\ 0 & \text{, otherwise} \end{cases}$$

Similarly, we can compute $P(v_k^l|\overline{\mathcal{R}}, L^k)$ following an approach based on a uniform p.d.f. over all possible values. These are essentially all the values in $V_{L^k}^{\mathcal{R}}(U)$ but only the values in $V_{L^k}^{\mathcal{R}}(U) \cap \overline{\mathcal{R}}_{L^k}(U)$ can be labeled either positive or negative (and we assume with the same probability) while the values in $V_{L^k}^{\mathcal{R}}(U) \setminus \overline{\mathcal{R}}_{L^k}(U)$ will surely be labeled negative. Therefore, it follows that $P(v_k^l|\overline{\mathcal{R}}, L^k) =$

$$\begin{cases} P(v_k^l|\overline{\mathcal{R}}, L^k) = \frac{1}{2\cdot|V_{L^k}^{\mathcal{R}}(U)\cap\overline{\mathcal{R}}_{L^k}(U)|} & \text{, iff } v_k \in V_{L^k}^{\mathcal{R}}(U) \cap \overline{\mathcal{R}}_{L^k}(U) \\ P(v_k^-|\overline{\mathcal{R}}, L^k) = \frac{1}{|V_{L^k}^{\mathcal{R}}(U)\setminus\overline{\mathcal{R}}_{L^k}(U)|} & \text{, iff } v_k \in V_{L^k}^{\mathcal{R}}(U) \setminus \overline{\mathcal{R}}_{L^k}(U) \\ 0 & \text{, iff } v_k \notin V_{L^k}^{\mathcal{R}}(U) \end{cases}$$

Note that the exact computation of the set $\overline{\mathcal{R}}_{L^k}(U)$ can be expensive, since given a value $v \in V_{L^k}^{\mathcal{R}}(U)$, in order to figure out whether $v \in \overline{\mathcal{R}}_{L^k}(U)$, we should enumerate a potentially very large number of vectors in $\overline{\mathcal{R}}_{L^k}(U)$.

We adopt an approximate and efficient solution based on the assumption that the equivalences holding for $k = 0$:[3] $V_{L^k}^{\mathcal{R}}(U), \overline{\mathcal{R}}_{L^k}(U) = V_{L^k}^{\mathcal{R}}(U)$ and $V_{L^k}^{\mathcal{R}}(U) \setminus \overline{\mathcal{R}}_{L^k}(U) = \emptyset$, also hold for any $k > 0$. Hence, it can be rewritten as:

$$P(v_k^l|\overline{\mathcal{R}}, L^k) \simeq \begin{cases} \frac{1}{2\cdot|V_{L^k}^{\mathcal{R}}(U)|} & \text{, iff } v_k \in V_{L^k}^{\mathcal{R}}(U) \\ 0 & \text{, iff } v_k \notin V_{L^k}^{\mathcal{R}}(U) \end{cases} \quad (3)$$

Actually, this is an oversimplification when k gets bigger and approaches $|U|$: both $\overline{\mathcal{R}}_{L^k}(U)$ and $V_{L^k}^{\mathcal{R}}(U)$ gets smaller and smaller and $V_{L^k}^{\mathcal{R}}(U) \setminus \overline{\mathcal{R}}_{L^k}(U) \neq \emptyset$. Since our algorithm look for the correct rule while minimizing k, in our setting this semplification does not significantly affect the results.

4.2 A-priori Probabilities

Our probabilistic model is based on the following *priors*: the prior probability $\mathcal{P}(\mathcal{R}^h)$ that a correct rule has been generated in the set of rules $\mathcal{R}^h$; the prior probability $\mathcal{P}(r)$ that the extraction rule r does extract the correct vector of values from the input set of pages U. As regards the former prior p.d.f. $\mathcal{P}(r)$, it is set by using a uniform p.d.f. over all rules in R_{L^1}.

For the latter priors $\mathcal{P}(\mathcal{R}^h)$, we follow a standard approach, and estimate the priors on a sufficiently large set of attributes the frequency of the involved events.

$\mathcal{P}(\mathcal{R}^h)$ has been set equals to $\frac{n_h}{N}$ where n_h is the number of attributes extracted by a rule in $\mathcal{R}^h$ and not in $\mathcal{R}^{h+1}$, and N is the number of attributes used in our estimation (we sampled $N = 356$ attributes).

[3]Admitting that every value is extracted at least by one rule.

Table 1: The main events considered in the bayesian analysis

Notation Probability	*Event*
$\mathcal{P}(r) = \mathcal{P}(r(U))$	prior probability that $r \in \mathcal{R}$ is correct
$\mathcal{P}(\mathcal{R})$ / $\mathcal{P}(\overline{\mathcal{R}})$	prior probability that a/none rule in $\mathcal{R}$ is correct
$P(r\|L^k)$	probability that r is correct once a l.s.s., observed L^k
$P(\overline{\mathcal{R}}\|L^k)$	probability that the correct rule is not in R, observed L^k
$P(v_k^l\|r, L^k)$	likelihood of v_k^l if r is correct, observed L^k
$P(v_k^l\|\overline{\mathcal{R}}, L^k)$	likelihood of v_k^l if the correct rule in not in R_{L^k}, observed L^k
$P(L^{k+1}) = P(v_k^{l_k}, L^k)$	probability of a l.s.s L^{k+1}

For example, 95% of the attributes can be extracted by rules in $\mathcal{R}^3$, while to cover the remaining 5% of attributes we should stretch our set of candidate rules to $\mathcal{R}^{10}$.

5. LEARNING EXTRACTION RULES

The probabilistic model just developed aims at at computing, observed an l.s.s. L^{k+1}, the probability $P(r|L^{k+1})$ that a given extraction rule r within a set of candidate rules $\mathcal{R}$ is correct, and the probability $P(\overline{\mathcal{R}}_{L^{k+1}})$ that the correct rule has not been generated at all, i.e. it is not inside $\mathcal{R}$.

Here, we start presenting an active learning algorithm that exploits these probabilities (Sec. 5.1); then, we describe a strategy to choose the next sample to be labeled (Sec. 5.2); finally, we discuss how the set of candidate rules is *dynamically* expanded according to the "observed need of expressiveness" rather than statically predetermined (Sec. 5.3).

5.1 Active Learning Algorithm

As shown by the pseudo-code in Listing 1, our algorithm takes as input a labeled sample sequence L built by actively [12] asking to an oracle (here modeled by means of the subprogram ORACLE()) the label of a sample chosen by the subprogram CHOOSESAMPLE(). The algorithm is parametric wrt a class of extraction rules decomposed into a hierarchy of subclasses $\{\mathcal{R}^h\}$, and it makes use of the probabilistic model detailed before (lines 8-9).

Initially, $\mathcal{R}^0$ is taken as initial set of candidate rules, and the set of rules admissible wrt the initial sample $\mathcal{R}^0_{L^1}$ is computed (lines 1-2). In every iteration, the oracle is asked to label a new sample v_k (lines 4-5) and the l.s.s. is updated to obtain L_{k+1} (6). Then, the set of admissible rules is updated (7) (recall that $R_{L^{k+1}} \subseteq R_{L^k}$), and the probabilities $P(r|L^{k+1})$ and $P(\overline{\mathcal{R}}_{L^{k+1}})$ are consequently updated (8-9). EXPANDRULESET() has to decide whether the set of candidate rule should be expanded (line 10).

This algorithm can be instantiated by appropriately choosing the semantics of three subprograms: CHOOSESAMPLE(), which selects the next sample to be labeled for the user; HALT(), which establishes an exit criterion before the l.s.s. naturally expires (i.e. R becomes empty); and finally, EXPANDRULESET(), which decides at runtime whether $\mathcal{R}^h$ should be expanded with new candidate rules by incrementing h. The latter decision is usually based on $P(\overline{\mathcal{R}}^h_{L^{k+1}})$: the higher its value, the more likely that new candidate rules are needed. The implementation of HALT() strongly depends on the overall goal of the search strategy. We leave further investigation on this aspect to future work; in the present paper, for illustrative purposes, we use a minimum threshold λ_r on the probability of the best rule:

HALT(R, L) { **return** $(\text{argmax}_{r \in R_L} P(r|L) > \lambda_r)$; }

Listing 1 An active learning algorithms for extraction rules

Input: set of sample pages $U = \{p_1, \ldots, p_{|U|}\}$
Input: an initial positive sample $L^1 = \{v_o^+\}$

Parameter: a hierarchy of *generable* rules $\{\mathcal{R}^h\}$ over U

Output: $P(r|L^{k+1})$ over $r \in \mathcal{R}^h_{L^{k+1}}$, $P(\overline{\mathcal{R}}^h_{L^{k+1}})$;

1: **let** $k \leftarrow 1$; **let** $h \leftarrow 0$;
2: **let** $R \leftarrow \mathcal{R}^h; R \leftarrow R_{L^1}$;
3: **while** $(R \neq \emptyset$ **and not** HALT$(R, L^k))$ **do**
4: $\quad v_k \leftarrow$ CHOOSESAMPLE(R, L^k);
5: $\quad l \leftarrow$ ORACLE(v_k);
6: $\quad L^{k+1} \leftarrow L^k \cup \{v_k^l\}$;
7: $\quad R \leftarrow \mathcal{R}^h_{L^{k+1}}$;
8: $\quad$ compute $P(r|L^{k+1})$, $\forall r \in R$ according to eq. 1;
9: $\quad$ compute $P(\overline{\mathcal{R}}^h|L^{k+1})$ according to eq. 2;
10: $\quad h \leftarrow h +$ EXPANDRULESET(R, L^{k+1});
11: $\quad k \leftarrow k + 1$;
12: **end while**
13: **if** $(R \neq \emptyset)$ **then**
14: $\quad$ **return** $\mathcal{R}^h_{L^{k+1}}$, $P(r|L^{k+1})$ and $P(\overline{\mathcal{R}}^h|L^{k+1})$;
15: **end if**
16: **return** $\perp$;

In the following, we detail CHOOSESAMPLE() and EXPANDRULESET(), respectively.

5.2 Choosing the Right Samples

For instantiating CHOOSESAMPLE() we propose two variants: ENTROPY plus a baseline algorithm RANDOM.

RANDOM: It chooses a random admissible sample:

CHOOSESAMPLE(R,L) { **return** a random $v \in V_L^R(U)$; }

and it serves two purposes: as a baseline against other strategies, and as a measure of the *sample complexity* of its extraction.

ENTROPY: It bases the sample choice on the p.d.f. of the extracted value: a simple strategy is to choose the sample on which rules most disagree, appropriately weighted according to their probability. This is equivalent to compute the *vote entropy* [12] for each $v \in R_{L^k}(U)$:

$$H(v) = -[P(v^+|L^k)\log P(v^+|L^k) + P(v^-|L^k)\log P(v^-|L^k)] \quad (4)$$

where: $P(v^+|L^k) = \sum_{r \in \{r \in R_{L^k} : r(p_v) = v\}} P(r|L^k)$

and: $P(v^-|L^k) = \sum_{r \in \{r \in R_{L^k} : r(p_v) \neq v\}} P(r|L^k)$

are the probabilities that v is respectively either a value to extract or an incorrect value.

The next sample is the one maximizing the vote entropy:

chooseSample(R,L) { **return** $\text{argmax}_{v \in V_L^R(U)} H(v)$; }

Note that this choice essentially removes the most uncertain sample.

5.3 Dynamically Expanding the Rule Set

expandRuleSet() is in charge of deciding whether and when expanding the hierarchy of candidate rules $\mathcal{R}^h$. It makes use of the probability $P(\overline{\mathcal{R}}^h_{L^k})$ of the correct rule not being present in the current set of candidate rules $\mathcal{R}^h$ after observing as input a given l.s.s. L^k.

We leave a thoroughly discussion of the best criteria underlying this analysis to future work, while in this paper we use a trivial implementation expandRuleSet() based on a predefined fixed threshold $\lambda_{\overline{\mathcal{R}}}$ over $P(\overline{\mathcal{R}}_{L^k})$:

```
expandRuleSet(R, L) {
  if (R = R^m) return 0; // max expansion reached
  else if (P(R̄_L) > λ_R̄) return +1;
  else return 0;
}
```

The set of rules is therefore enlarged *lazily*, i.e. only whenever according to $P(\overline{R}_L)$ there is evidence that a correct rule is not amongst the currently available candidate rules.

6. EXPERIMENTS AND FUTURE WORKS

We built a Java prototype implementation of our algorithms.We report the results of some preliminary experiments mainly focused on evaluating the effectiveness of the SRM technique. We downloaded pages from 101 websites publishing information in several domain. For each website, we downloaded a small set of sample pages (about 20) sharing a common HTML template, and considered as relevant $2-4$ attributes, for which we manually crafted a *golden* extraction rule to get as a reference the correct values, totally considering 240 attributes.

The results have been collected running our prototype with $\lambda_r = 0.99$ and $\lambda_{\overline{\mathcal{R}}} = 0.95$. Table 6 reports: the (rounded) average number of MQ by Random over 10 executions (*Rnd*);[4] the aggregated number of attributes extracted by Random ($\#A$); the number of MQ posed by Entropy with $\mathcal{R}^5$ and SRM disabled, (H); the number of MQ when SRM starts from the class of extraction rules $\mathcal{R}^i$ with $i = 0, 1, 2, 3$ (*SRM*-$\mathcal{R}^i$). The last row reports the average precision of the values extracted by the output rules.

SRM always saves MQ. Whenever the initial expressiveness is low, the precision suffers, and the number of MQ saved is higher. This can be explained by expandRuleSet making wrong decisions, i.e. it trusts on current imprecise rules rather than betting on more expressive classes. Note that even Entropy cannot reach 100% since there exist attributes that need rules out of $\mathcal{R}^5$.

Since our dataset consists of a small number of pages per site and it is not suitable for evaluating the sampling issue, we leave it to future work. To overcome the precision loss with the SRM approach, it is possible to add additional informations, e.g. types analysis (to weight more vectors with homogeneous type). Also, we aim at integrating SRM with automatic annotators that rely on data available on external sources, such as Freebase, and with crowdsourcing platforms.

[4]These values represent the sample complexity.

Static			*SRM*			
Rnd	#A	H	$\mathcal{R}^0$	$\mathcal{R}^1$	$\mathcal{R}^2$	$\mathcal{R}^3$
≥ 9	10	4	1.90	2.33	2.89	3.56
8	16	4.12	1.56	1.78	3.22	3.25
7	12	4.08	1.66	1.69	2.7	3.63
6	15	3.86	1.26	1.58	3.22	3.40
5	15	3.5	1.55	2.06	2.39	2.77
4	32	3.58	1.46	1.96	2.48	3.13
3	53	3.45	1.55	2.06	2.52	3.14
2	55	3.01	1.76	1.54	2.11	2.38
1	128	2.27	1.59	1.65	1.71	1.76
Precision		96.2%	86.7%	93.79%	95.85%	96.2%

Table 2: Experimental results.

7. REFERENCES

[1] D. Angluin. Queries revisited. *Theor. Comput. Sci.*, 313(2):175–194, 2004.

[2] A. Arasu and H. Garcia-Molina. Extracting structured data from web pages. In *SIGMOD* 2003.

[3] M.-F. Balcan, S. Hanneke, and J. W. Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3), 2010.

[4] C.-H. Chang and S.-C. Lui. IEPAD: information extraction based on pattern discovery. In *WWW* 2001.

[5] V. Crescenzi and G. Mecca. Automatic information extraction from large websites. *J. ACM*, 51(5):731–779, 2004.

[6] V. Crescenzi and P. Merialdo. Wrapper inference for ambiguous web pages. *JAAI*, 22(1&2):21–52, 2008.

[7] N. N. Dalvi, R. Kumar, and M. A. Soliman. Automatic wrappers for large scale web extraction. *PVLDB*, 4(4):219–230, 2011.

[8] T. Furche, G. Gottlob, G. Grasso, O. Gunes, X. Guo, A. Kravchenko, G. Orsi, C. Schallhart, A. J. Sellers, and C. Wang. Diadem: domain-centric, intelligent, automated data extraction methodology. In *WWW (Companion Volume)* 2012.

[9] G. Gottlob, C. Koch, R. Baumgartner, M. Herzog, and S. Flesca. The lixto data extraction project - back and forth between theory and practice. In *PODS* 2004.

[10] U. Irmak and T. Suel. Interactive wrapper generation with minimal user effort. In *WWW* 2006.

[11] I. Muslea, S. Minton, and C. A. Knoblock. Active learning with multiple views. JAIR, 27:203–233, 2006.

[12] B. Settles. Active learning literature survey. CS T.R. 1648, Univ. of Wisconsin–Madison, 2009.

[13] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

[14] V. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[15] Y. Zhai and B. Liu. Structured data extraction from the web based on partial tree alignment. *IEEE Trans. Knowl. Data Eng.*, 18(12):1614–1628, 2006.