Web Information Extraction Using Markov Logic Networks

Sandeepkumar Satpal [†1] Sahely Bhadra [#2] S Sundararajan [*3] Rajeev Rastogi [*4] Prithviraj Sen [*5]

[†]Microsoft Hyderabad India
[*]Yahoo! Labs Bangalore India
[#]CSA, Indian Institute of Science Bangalore India

[1]ssatpal@microsoft.com,{[3]ssrajan, [4]rrastogi, [5]sen}@yahoo-inc.com, [2]sahely@csa.iisc.ernet.in

ABSTRACT

In this paper, we consider the problem of extracting structured data from web pages taking into account both the content of individual attributes as well as the structure of pages and sites. We use *Markov Logic Networks* (MLNs) to capture both content and structural features in a single unified framework, and this enables us to perform more accurate inference. We show that inference in our information extraction scenario reduces to solving an instance of the *maximum weight subgraph* problem. We develop specialized procedures for solving the maximum subgraph variants that are far more efficient than previously proposed inference methods for MLNs that solve variants of MAX-SAT. Experiments with real-life datasets demonstrate the effectiveness of our approach.

Categories and Subject Descriptors: I.7m [Computing Methodologies]: Document and Text Processing - Miscellaneous
General Terms: Algorithms, Design
Keywords: Information extraction, Markov logic networks

1. INTRODUCTION

Extracting structured data from web pages is a key challenge due to the presence of noise, and we focus on this problem in this paper. Consider a set of web sites $\mathbf{W}$ belonging to a specific domain like Restaurants, Books, etc. We denote the set of attributes (e.g.,Name, Address, Price and Phone) that we want to extract from the web sites in $\mathbf{W}$ by $\mathbf{A}$. For each web page $p \in W \in \mathbf{W}$, consider the DOM tree representation of the page. Then the extraction problem is to assign *attribute labels* to all the leaf nodes in the DOM tree for p.

In recent years, there has been a flurry of research activity on statistical models for information extraction [2]. These models leverage both content and structural cues to detect attribute values, and are tolerant to errors and noise in the input pages. For extracting data from the web with little human supervision, *Hierarchical Conditional Random Fields* (HCRFs) [2] represent the state-of-the-art today. Despite their expressive power, a limitation of HCRFs is that they only capture short-range structural dependencies between neighboring elements in the DOM tree. Here, we develop an MLN-based framework for general-purpose extraction from web sites.

2. OUR APPROACH

Our MLN models capture both attribute content properties as well as long-range structural relationships in a single unified framework, and this enables us to perform extractions with higher accuracy. Markov Logic Network (MLN) [1] is a set of pairs (F_i, w_i), where F_i is a first-order formula and w_i is its corresponding weight. Now, for a web site W, let x be the set of evidence predicates that are true for pages in W. Then, the probability that the set of query predicates q is true is given by: $P(q|x) = \frac{1}{Z}\exp(\sum_{F_i}\sum_{g\in G_i} w_i \cdot g(q \cup x))$ where G_i is the set of groundings of F_i, $g(q \cup x)$ equals to 1 if the grounded formula g is true for predicate set $q \cup x$ and 0 otherwise, and Z is a normalization constant which ensures that probabilities add up to 1.

Predicates and Formulas: Content formulas link content predicates (e.g.,$\mathsf{Has5Digits}(n)$) with query predicates for a node n. For each content predicate, attribute pair, C_i, A_j, we create a separate content formula $\forall n\ C_i(n) \Rightarrow \mathsf{IsA_j}(n)$. Structural formulas connect structural predicates and non-noise query predicates. Structural predicates listed below capture such long-range relationships over more than one node either within the same page or across the pages of a site. (1) *Proximity:* The proximity predicate $\mathsf{Close}(n_1, n_2)$ reflects the *closeness* between a pair of nodes n_1, n_2 in a page p and proximity formulas have the general form $\forall n_1, n_2 \in p\ \mathsf{IsA_i}(n_1) \wedge \mathsf{IsA_j}(n_2) \Rightarrow \mathsf{Close}(n_1, n_2)$. (2) *Precedence:* Precedence formulas specify *ordering* relationships among the attributes. Such ordering can be captured using precedence formulas like $\forall n_1, n_2 \in p\ \mathsf{IsName}(n_1) \wedge \mathsf{IsNumberofPages}(n_2) \Rightarrow \mathsf{Precedes}(n_1, n_2)$. (3) *Alignment:* The structural predicate $\mathsf{Aligned}(n_1, n_2)$ captures the similarity in non-noise attribute value positions across the template based pages of a web site. We consider a pair of nodes n_1, n_2 in two different pages p_1, p_2 to be aligned (that is, $\mathsf{Aligned}(n_1, n_2)$ is true) if n_1 and n_2 have identical DOM paths. An example is $\forall n_1 \in p_1, n_2 \in p_2\ \mathsf{IsName}(n_1) \wedge \mathsf{IsName}(n_2) \Rightarrow \mathsf{Aligned}(n_1, n_2)$.

Learning and Inference: Given the formulas F_i in our MLN model, the learning problem is to learn the weights w_i. We find these weights by maximizing a regularized log likelihood function [1] defined over the training set. Our data extraction problem is to find the most likely attribute label assignment for page nodes. This is equivalent to finding the assignment q^* that maximizes the sum of weighted formulas given by: $\sum_{F_i}\sum_{g\in G_i} w_i \cdot g(q)$ (for simplicity, we drop the dependence of x in $g(q \cup x)$). Thus, the inference problem is an instance of the weighted MAX-SAT problem, which is known to be NP-hard [1]. Since the well-known MAXWALKSAT (MWS) algorithm is inefficient for our purpose, we devise efficient graph-based inference algorithms.

3. INFERENCE ALGORITHM

Our graph framework leverages the fact that soft ground formulas g contain one or two nodes. Content formulas contain only one node while structural formulas like proximity or alignment are defined over two nodes. For a node n_i, let G_{n_i} denote the set of content ground formulas containing node n_i, and let G_{n_i,n_k} denote the structural ground formulas containing nodes n_i and n_k. Then, our optimization problem is to find a q that satisfies some hard constraints (as illustrated below) and maximizes the function:

$$X(q) = \sum_{n_i} \sum_{g \in G_{n_i}} w(g) \cdot g(q) + \sum_{n_i,n_k} \sum_{g \in G_{n_i,n_k}} w(g) \cdot g(q) \quad (1)$$

Above, for grounding g derived from formula F_i, the weight $w(g)$ is equal to w_i. Now, we show that finding a q that maximizes Equation (1) is equivalent to computing the maximum weight subgraph in a graph $\mathcal{G}$ constructed as described below. In $\mathcal{G}$, there is a separate vertex v_{ij} for each node n_i within a page and each attribute A_j. Thus, there are $|\mathbf{A}|$ layers of vertices in $\mathcal{G}$ with layer j corresponding to attribute A_j. Intuitively, vertex v_{ij} in $\mathcal{G}$ corresponds to the predicate $\mathsf{IsA_j}(n_i)$ in q. With each vertex v_{ij}, we associate a weight $t(v_{ij}) = \sum_{g \in G_{n_i}} w(g) \cdot g(\{\mathsf{IsA_j}(n_i)\})$. To incorporate the contribution of distinct node pairs n_i, n_k to $X(q)$, we set the edge weight $t(v_{ij}, v_{kl})$ to $\sum_{g \in G_{n_i,n_k}} w(g) \cdot g(\{\mathsf{IsA_j}(n_i), \mathsf{IsA_l}(n_k)\})$. Now, for query predicates q, consider the subgraph of $\mathcal{G}$ containing vertices v_{ij} for each predicate $\mathsf{IsA_j}(n_i) \in q$. It is easy to see that the sum of the weights of vertices and edges in this subgraph induced by q is equal to $X(q)$. Thus, the problem of computing a label assignment q that maximizes $X(q)$ is equivalent to finding the maximum weight subgraph in $\mathcal{G}$. Note that in any label assignment, a majority of the nodes will be labeled as Noise. So we improve the efficiency of our inference algorithm by considering Noise as the "default" label, and only computing the non-noise label assignments in a modified graph $\hat{\mathcal{G}}$.

Algorithm 1 Maximum weight subgraph computation.

```
Input: Graph $\hat{\mathcal{G}}$.
Output: Maximum weight subgraph of $\hat{\mathcal{G}}$.

$S = \emptyset$;            ($S$ represents the set of nodes in the subgraph)
repeat
  $S_{old} = S$;
  Let $v_{ij}$ be the vertex for whom $\hat{t}(S \cup \{v_{ij}\})$ is maximum;
  if $\hat{t}(S \cup \{v_{ij}\}) > \hat{t}(S)$ then $S = S \cup \{v_{ij}\}$;
until $\hat{t}(S) = \hat{t}(S_{old})$        ($\hat{t}(S)$ represents the weight of $S$)
repeat
  $S_{old} = S$;
  Consider a random ordering of (page, attribute) pairs;
  for each $(p_r, A_j)$ pair in the ordering do
    if $S$ contains a vertex $v_{i,j}$ from layer $j$ of page $p_r$ then
      $S' = S - \{v_{ij}\}$;
      Let $v_{kj}$ be the vertex in layer $j$ of page $p_r$ for whom
      $\hat{t}(S' \cup \{v_{kj}\})$ is maximum;
      if $\hat{t}(S' \cup \{v_{kj}\}) > \hat{t}(S)$ then $S = S' \cup \{v_{kj}\}$;
      if $\hat{t}(S') > \hat{t}(S)$ and $A_j$ is non-mandatory then $S = S'$;
    else
      Let $v_{ij}$ be the node in layer $j$ of page $p_r$ for whom $\hat{t}(S \cup$
      $\{v_{ij}\})$ is maximum;
      if $\hat{t}(S \cup \{v_{ij}\}) > \hat{t}(S)$ or $A_j$ is mandatory then
        $S = S \cup \{v_{ij}\}$;
    end if
  end for
until $\hat{t}(S) - \hat{t}(S_{old}) < \epsilon$  ($\epsilon$: a user-defined threshold parameter)
return $S$;
```

Enforcing Hard Constraints: We model the single label hard constraint of assigning exactly one attribute label to each node by assigning a weight of $-\infty$ to the edge between vertices v_{ij} and v_{il} in $\hat{\mathcal{G}}$ corresponding to the same node n_i, but in different layers j and l. We incorporate contiguity constraints (that is, nodes with the *same* attribute label are contiguous) by constructing super-vertices (coalesced subsequences of nodes) with suitably aggregated weights. Note that contiguity constraints involve 3 nodes and pose a serious challenge for satisfiability solvers.

Method	Name	Address	Phone	Timing	Description	Review	Avg-F1
C	**94.9**	**96.8**	91.9	70.8	66.3	**90.2**	85.2
C+S (Intra)	**94.9**	95.6	95.9	91.3	65.9	90.2	89.0
C+S (All)	94.6	96.7	**96.6**	**92.6**	**73.1**	86.8	**90.1**
MWS (WOC)	83.7	66.8	92.4	76.9	53.0	39.7	68.7

Table 1: Average F1 scores on the Restaurants dataset.

A Greedy Algorithm: Algorithm 1 describes a greedy algorithm to find the maximum weight subgraph S in $\hat{\mathcal{G}}$. Note that nodes n_i with corresponding vertices v_{ij} in S are labeled with the attribute A_j; the remaining nodes are labeled as Noise.

4. EXPERIMENTAL EVALUATION

Setup: We constructed two datasets Restaurants and Books by picking random web pages from five popular websites for each domain. The approximate number of pages we collected per site were 100 and 500 for Restaurants and Books, respectively. To learn the MLN model, we made use of the publicly available package *thebeast* (`http://code.google.com/p/thebeast/`). All the experiments were performed in a leave-one-out (LOO) setting (at website level). Due to space limitation we present results from only one dataset.

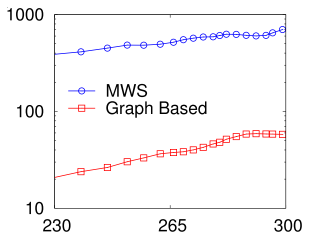

Figure 1: Inference time in seconds versus Number of nodes for C+S (All) on the Restaurants dataset.

The first three rows of Table 1 depict the F1 values for the MLN model using our inference algorithm. We observe that the F1 performance improves for several attributes as we go from the model with only content formulas (C) to the model that includes intra-page (C+S (Intra)) and inter-page structural formulas (C + S (All)) (with 3 pages in a group).

Due to speed and memory issues, we could not obtain F1 score results for the MWS algorithm with contiguity constraints. Therefore, the MWS results reported in the final row of Table 1 are for the case of C+S (Intra) and without contiguity constraints (WOC), with the number of random flips set to 80000 which gave the maximum labeling accuracy. Although the MWS algorithm ran fast in this case, its F1 performance was poor. In Figure 1, we plot the running times for page groups (C+S(All)) containing different numbers of nodes. To ensure a certain minimum level of labeling accuracy, we set the number of flips to 1000. It is easy to see that our algorithm is an order of magnitude faster.

We observed that while the HCRF model generalizes well within the sites, its generalization is not good on unseen sites. In contrast, the MLN model with structural formulas and contiguity constraints (tested under the more stringent LOO setting) performs significantly better (more than 25%).

5. REFERENCES

[1] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[2] J. Zhu, Z. Nie, J. Wen, B. Zhang, and W. Ma. Simultaneous record detection and attribute labeling in web data extraction. In *ACM SIGKDD*, 2006.