Bootstrapping Semantic Annotation for Content-Rich HTML Documents

Saikat Mukherjee I.V. Ramakrishnan Amarjeet Singh
Department of Computer Science
Stony Brook University,
Stony Brook, NY 11794, U.S.A.
{saikat,ram,amarjeet}@cs.sunysb.edu

Abstract

Enormous amount of semantic data is still being encoded in HTML documents. Identifying and annotating the semantic concepts implicit in such documents makes them directly amenable for Semantic Web processing. In this paper we describe a highly automated technique for annotating HTML documents, especially template-based content-rich documents, containing many different semantic concepts per document. Starting with a (small) seed of hand-labeled instances of semantic concepts in a set of HTML documents we bootstrap an annotation process that automatically identifies unlabeled concept instances present in other documents. The bootstrapping technique exploits the observation that semantically related items in content-rich documents exhibit consistency in presentation style and spatial locality to learn a statistical model for accurately identifying different semantic concepts in HTML documents drawn from a variety of Web sources. We also present experimental results on the effectiveness of the technique.

1 Introduction

Semantic Web envisions a next-generation information network where content providers define and share machine processable data on the Web. A primary aspect of Semantic Web documents is that they contain metadata to express the meaning of their content. But an enormous amount of extant semantic data (such as product descriptions and pricing information, different categories of news, etc) is still being encoded in "plain" HTML documents. Although RDF/XML has been widely recognized as the standard vehicle for representing semantic information on the Web, we can extend the reach of Semantic Web to HTML documents by identifying and annotating the (implicit) semantic concepts that are present in their content.

Early solutions [15, 10] to this problem were based on hand-crafted ontologies and graphical ontology/annotation editors that facilitated manual mapping of unlabeled document segments to ontological concepts. From an automation standpoint they are at the "low-degree-of-automation" end of the solution spectrum. The technique in [7] as well as our previous work [22] cover the middle ground wherein document segmentation is done automatically and assignment of semantic labels to these segments is done with manually-crafted ontologies and knowledge bases.

Such a labeling process, via hand-crafted ontologies, is expensive and time-consuming. This begs the question: *Can this semantic labeling step be automated?* We address this question in this paper. Specifically we couple our structural analysis technique, described in [22], for partitioning Web pages into unlabeled segments (consisting of "semantically related" items) with a highly automated labeling step, to provide a "high-degree-of-automation" solution to the problem of semantic annotation of HTML documents.

Overview of our Approach: Our approach to structural analysis was based on the simple but useful observation that semantically related items in a HTML page normally exhibit *consistency in presentation style* and *spatial locality*. This is particularly true of content-rich Web sites that update frequently such as news portals, education and e-commerce sites, because these sites are typically maintained using content management software that create HTML pages by populating templates from backend databases.

Figure 1(a) and (b) are two examples of such Web sites. Note the consistency in presentation style of items in the news taxonomy in the left corner in Figure 1(a). The main taxonomic items, "NEWS", "OPINION", "FEATURES", ..., etc., are all presented in bold font. All the subtaxonomic items (*e.g.*, "International", "National", "Washington", ..., etc.) under a main taxonomic item (*e.g.*, "NEWS") are hyperlinks. This kind of consistency in presentation style has a very strong manifestation in the Document Object Model (DOM) tree of an HTML document. For example, Figure 2(a) is a fragment of the DOM tree for New York Times home page shown in Figure 1(a). The root-to-

Proceedings of the 21st International Conference on Data Engineering (ICDE 2005)
1084-4627/05 $20.00 © 2005 **IEEE**

(a) (b)

Figure 1. (a) New York Times home page (b) Washington Post home page

leaf sequences of HTML tags for the nodes "NEWS" and "FEATURES" are exactly the same, as are the sequences of HTML tags for the nodes "International", "Arts", etc. (font tags with different attributes, *e.g.*, size, are distinguished using different subscripts in Figure 2(a)).

Spatial locality in a HTML page and its corresponding DOM tree can also indicate content similarity. For example, when rendered in a browser all the taxonomic items in New York Times are placed in close vicinity occupying the left portion of the page (see Figure 1(a)). In the corresponding DOM tree all these taxonomic items are grouped together under one *single* subtree rooted at the *table* node (see Figure 2(a)). Similarly, all the major headline news items are clustered under a different subtree rooted at the *td* node (shown circled in Figure 2(a)). The structural analysis technique that we had developed in [22] groups together elements in a HTML page into an unlabeled tree of partitions consisting of semantically related items (see Figure 2(b)).

Observe from Figure 1(a) and (b) that there is consistency in presentation of semantic concepts even among documents drawn from different sources. For example, the taxonomy news concept in both New York Times and Washington Post are characterized by the words "NEWS", "OPINION", "Business", etc. Moreover the presentation style of this concept, with taxonomic items (such as "NEWS") appearing as text and sub-taxonomic items (such as "National") appearing as hyperlinks, is also consistent in both the pages. The implication is that such consistent presentation styles exhibit discerning *features* of a concept that can facilitate its identification in unlabeled HTML pages.

A consequence of spatial locality is that semantically related content elements appear under a common node in the unlabeled partition tree (e.g. the "group" node, enclosed within the solid circle in Figure 2(b), corresponding to taxonomy news). The degree of similarity between the content in the subtree rooted at the common node and those rooted at its children is "higher" than those rooted at its siblings. (e.g. the group node, enclosed within the dashed circle in Figure 2(b), corresponding to major headlines news). We can exploit this observation to identify concept instances with precision.

In this paper we use the above ideas and observations to develop a highly automated statistical-based algorithm for identifying concept instances in content-rich HTML documents. The input to the algorithm is a (small) set of hand-labeled concept instances from HTML pages. Based on this "seed" the algorithm bootstraps an annotation process that automatically recognizes unlabeled concept instances in new HTML pages and assigns semantic labels to them. Our annotation process will generate the semantic partition tree (Figure 2(c)) from the structural partition tree (Figure 2(b)). An *important aspect* of our algorithm is that it does not use hand-crafted ontologies.

Although the research literature is rife with techniques for document classification (see [28] for a text classification survey) they are not directly applicable to our problem. This is because document classification addresses the problem of classifying the content of the entire document to one concept, whereas our semantic annotation problem involves identifying multiple concepts within the same document.

Summary of contributions:

- Based on the observation that semantic concepts exhibit consistency in presentation style, we exploit the presentational aspects of HTML documents to learn highly discernible concept features (Section 2.2).
- We develop a Bayesian framework to incorporate these features into a statistical model of concepts. We use

this model to compute a likelihood measure for a concept at every node in the partition tree (Section 2.3). The likelihood measure of a node for a concept is an estimate of the "closeness" of the content in the subtree rooted at that node to the concept.

- We use the likelihood estimates in conjunction with the spatial locality observation to assign semantic labels to nodes in the partition tree. Nevertheless, overlapping features amongst instances of different concepts can cause ambiguity in concept identification. We develop a novel bipartite-graph based ambiguity resolution technique to facilitate disambiguation in such cases and thereby improve the precision of semantic label assignment (Section 2.4).

The rest of this paper is organized as follows: Section 2 presents the technical details of our semantic annotation framework. Section 3 presents experimental results while related work and discussions appear in Section 4 and Section 5 respectively.

2 Semantic Annotation

2.1 Structural Analysis

The essence of the idea underlying the construction of the unlabeled partition tree by structural analysis is this: Consistency in presentation style and spatial locality of semantically related items in HTML documents is identified by looking for recurring patterns in the path structures of the corresponding DOM trees. For example, the root-to-leaf path strings of the news taxonomy items, such as "NEWS", "OPINION", etc., in Figure 1(a), which consist of tag names and their associated attributes (see Figure 2(a)), are all identical; let us denote these identical strings using T_1. Similarly, the root-to-leaf path strings of the news subtaxonomy items, such as all the links under the "NEWS" item ("International", "National", etc.) are also the same; let us denote it using T_2. Then clearly we can see that the pattern T_1T_2* captures the structural recurrence of the subtree rooted at $table$ (shown in circle) in Figure 2(a). By this means, discovery of semantically related items can be achieved by mining sequential patterns. Algorithmic details of pattern discovery for structural analysis appear in our earlier work in [22].

We create a *group* node in the partition tree corresponding to the set of recurring patterns discovered under a node in the DOM tree (e.g. the group node enclosed within the solid circle in Figure 2(b) corresponds to the repeating pattern T_1T_2* discovered under the *table* node in Figure 2(a)). In addition we create a *pattern* node for each instance of the pattern in the repeating sequence. They become the children of the group node (e.g. see the pattern nodes enclosed within solid circles in Fig. 2(b). Each of these pattern nodes corresponds to the pattern $T_1T_2T_2 \ldots T_2$ with T_2 occurring zero or more times in it).

The annotation problem that we address in this paper is this: We are given a collection of HTML pages and a corresponding set of unlabeled partition trees constructed from these documents – a tree per HTML page. A subset of the nodes in the partition trees have been manually labeled as concept instances such as the circled node in Fig. 2(b). Develop a method that uses these labeled examples to automatically identify and annotate all the other nodes in all the partition trees that correspond to concept instances.

2.2 Feature Extraction

From these labeled examples our goal is to learn a statistical model of concept features. The learned model will be used for identifying unlabeled concept instances in Web pages. The first task at hand is to extract features from these examples. The feature space is drawn from both the content of the partitions as well as the style with which the content is presented. Given a node p_k in the partition tree, feature extraction generates a list of $\langle f_i, n_{f_i,p_k} \rangle$ pairs, where n_{f_i,p_k} is the weight of feature f_i in p_k. In our development of feature extraction we divide the feature space into two broad categories, namely, *unstructured* and *structured* features described as follows.

Unstructured Features: After eliminating stop-words the bag of words in the partition tree constitute the unstructured elements in the feature space. Each feature element is assigned a weight at every node in the partition tree. To understand how weights are assigned we make a few observations.

Usually the labels of (small) partitions deep in a partition tree are provided by Web site designers in the document itself (*e.g.*, "BUSINESS," "NATIONAL" ..., etc. appearing in the third column in Figure 1(a) which are instances of the category news concept). We should assign a relatively higher weight to such words since they are in some sense the "constant" features of the template. When such a label is present in a document, it is usually the first item in the partition; the remaining items are all semantically related. When constructing the partition tree these remaining items become children of a group node $p_{i'}$ and the first item $p_{i''}$ becomes the sibling of $p_{i'}$. Together they appear as the children of a pattern node p_i. (See illustration of this process in Figure 2(b) for taxonomy news). Taking these weight impacting factors into account we use the following function to assign weights to an unstructured feature f_i at a node p_k having λ_{p_k} children:

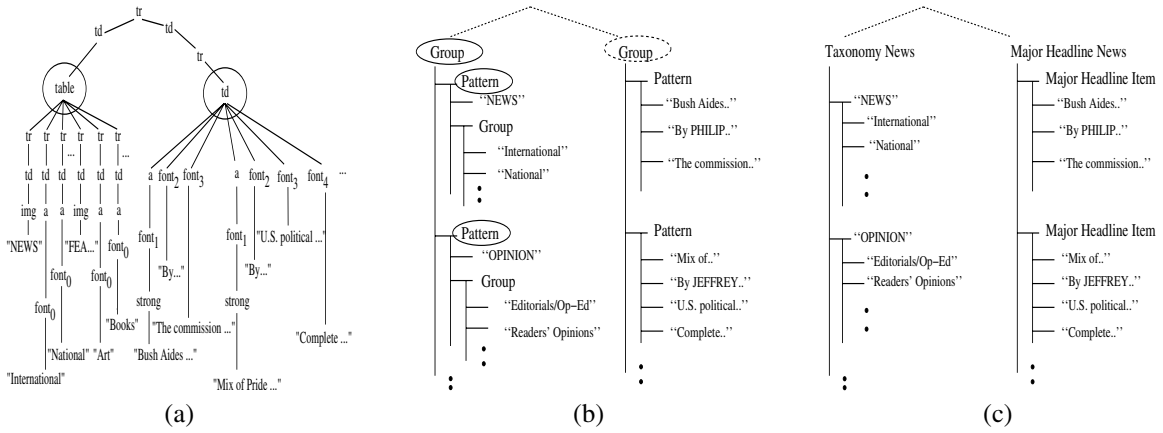

Figure 2. (a) DOM fragment of the New York Times home page (b) Unlabeled Partition tree of the corresponding fragment (c) Semantic Concept Tree of the corresponding fragment

$$n_{f_i,p_k} = \begin{cases} n_{f_i,p_{k'}} + \lambda_{p_{k'}} \times n_{f_i,p_{k''}} & \text{, if } p_k \text{ has two children } p_{k'} \text{ and } p_{k''} \text{ and } p_{k'} \text{ is a group node} \\ \sum_{p_{k'}} n_{f_i,p_{k'}} & \text{, for all other internal nodes } p_k \\ \text{number of occurrences of } f_i & \text{, if } p_k \text{ is a leaf partition} \end{cases}$$

The summation in the second case ranges over all the children $p_{k'}$ of p_k. For instance, in the partition tree corresponding to the page in Figure 1(a), the feature "BUSINESS" with non-zero weight will be associated with a node that will be a sibling of the group node denoting the set of links "Dow prunes..", "Oil prices..", and "G.M..". The weight of the feature "BUSINESS" will be increased by the number of this group node's children (which is 3 in this case).

Structured Features: Whereas unstructured features represent important words that appear in the textual content of partitions, structured features capture the presentational aspects of their content. For instance, in Figure 1(a), each major headlines news item is presented as a link ("Bush Aides.."), followed by two consecutive text strings ("By PHILIP..", "The commission.."), and an optional link ("Complete.."). Abstractly speaking the presentation style is captured by the sequence: $link \cdot text \cdot text \cdot ?link$ where $?link$ means that this $link$ may not always be present in all headline news items (akin to the ? operator used in the language of regular expressions).

Formally we say that a $link$ element (denoting a hyperlink) and a $text$ element (denoting a text string) in a HTML document are *basic structural elements (bse)'s*. A *complex structural element (cse)* is a sequence of one or more *bse's*. The structured features in the feature space are *cse's*.

Just as we assigned a weight to unstructured features at every node we will also assign weights to structured features. Note that the structured feature of a leaf node is a *bse* which is either a $link$ type or $text$ type since leaf nodes in the partition tree contain either hyperlinks or text strings.[1] We propagate the structured features of the leaf nodes up the tree to construct the structured features of the internal nodes and assign weights to them. The structured features of internal nodes are a set of *cse's*. They are constructed thus: If an internal node is not a pattern node then its structured feature set is the union of it's children's structured features. The weight of each feature in this set is the cumulative sum of the feature's weight in each of the node's children. Recall that a pattern node denotes an instance of a repetitive pattern mined by structural analysis and it reflects the presentational style of semantically related elements. So the structured feature of a pattern node is obtained by concatenating the structured features of its children. Since we want to make a determination of concept instances using features that will always be present, features representing the optional aspect of the pattern are omitted.

Formally, if p_k is a node with $p_{k_1}, ..., p_{k_m}$ as its children then it's set of $\langle$ structured feature, weight $\rangle$ pairs, F_{pk}, is defined to be:

$$F_{p_k} = \begin{cases} \{\cup_{f_i} \langle f_i, \sum_{j=1}^{j=m} n_{f_i,p_{k_j}} \} & \text{, if } p_k \text{ is a non-pattern internal node and the union is over all } f_i \text{ in } p_{k_1}, ..., p_{k_m} \\ \{\langle f_{n_1} \cdot f_{n_2} \cdot ... \cdot f_{n_l}, 1 \rangle, \cup_{f_i} \langle f_i, \sum_{j=1}^{j=m} n_{f_i,p_{k_m}} \rangle\} & \text{, if } p_k \text{ is a pattern node and } f_{n_i} \text{ is a non-optional feature of } p_{k_i} \\ \{\langle link, 1 \rangle\} & \text{, if } p_k \text{ is a link leaf node} \\ \{\langle text, 1 \rangle\} & \text{, if } p_k \text{ is a text leaf node} \end{cases}$$

[1] We do not use other leaf elements such as images, etc. in our feature space

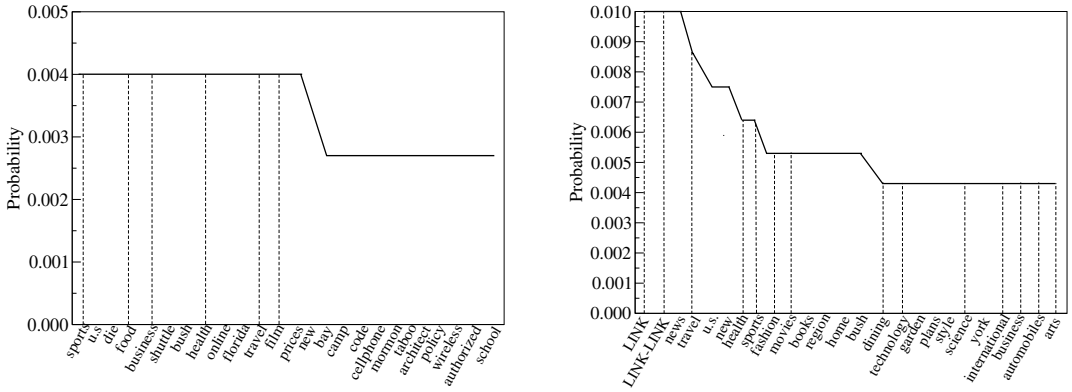

Figure 3. Top 25 Features and their Probabilities for the Category News concept model (a) with words (b) with structured and unstructured features

For instance, in Figure 2(b), the leaf partitions "Bush Aides..", "By PHILIP..", and "The commission.." have structured features $link$, $text$, and $text$ respectively. Similarly, the leaf partitions "Mix of..", "By JEFFREY..", "U.S. political..", and "Complete.." have the features $link$, $text$, $text$, and $link$ respectively. Structural analysis on the entire sequence of major headlines, shown in Figure 1(a), yields the set of structured features $\{\langle link \cdot text \cdot text, 1\rangle, \langle link, 1\rangle, \langle text, 2\rangle\}$ for the first pattern node. Similarly, the set of structured features for the second pattern node is $\{\langle link \cdot text \cdot text, 1\rangle, \langle link, 2\rangle, \langle text, 2\rangle\}$. Note the $link$ element denoting "Complete .." is optional and hence is discarded from the structured feature set of the 2nd pattern node. Finally, the set of structured features at the group node (considering these two pattern nodes only) is $\{\langle link \cdot text \cdot text, 2\rangle, \langle link, 3\rangle, \langle text, 4\rangle\}$.

2.3 Concept Model

We now develop a Bayesian model for concept identification based on the features defined above. This model will be learned from manually labeled examples of concept instances.

The model of a concept is a probabilistic distribution of both structured and unstructured features. This distribution is learned from the labeled example set. Recall that feature extraction from a node p_k in the partition tree yields the set of $\langle f_i, n_{f_i,p_k}\rangle$ pairs. Given a set L of labeled nodes that are instances of concept c_j, the probability of occurrence of a feature f_i in c_j is defined using Laplace smoothing as:

$$P(f_i|c_j) = \frac{\sum_{k=1}^{k=L} n_{f_i,p_k} + 1}{\sum_{i=1}^{i=|F_{train}|} \sum_{k=1}^{k=L} n_{f_i,p_k} + |F_{train}|}$$

F_{train} denotes the set of features present in the training instances in L.

Figure 3(b) illustrates the probabilistic distribution of structured and unstructured features for the category news concept model while Figure 3(a) shows the corresponding model using only words as features. In both cases, the model was trained from 2 labeled home pages, one from New York Times and the other from CNN . The features on which the dotted lines are anchored on the horizontal axis were determined to be important for the category news concept. Observe in Figure 3(b) that usage of structured/unstructured feature extraction results in identifying more relevant concept features than using a model based solely on words as features (Figure 3(a)).

We utilize $P(f_i|c_j)$ to compute the probability of a node, with a set of features F, being an an instance of a concept c_j. We use a Bayesian method. Specifically, by Bayes theorem,

$$P(c_j|F) = \frac{P(c_j) \times P(F|c_j)}{P(F)}$$

Assuming a uniform prior over concepts in any partition and ignoring the term $P(F)$, which is independent of any concept, computing $P(c_j|F)$ reduces to computing the likelihood $P(F|c_j)$. A multinomial distribution [2], that takes into account the weights of the features, is used to model the likelihood $P(F|c_j)$. Formally:

$$P(c_j|F) \propto \left(\frac{N_F!}{N_{f_1,p_k}! \ldots . N_{f_{|F|},p_k}!}\right) \times \prod_{i=1}^{i=|F|} P(f_i|c_j)^{N_{f_i,p_k}}$$

where N_F denotes a normalized number of features obtained by scaling $|F|$ to it and N_{f_i,p_k} denotes the scaled value of n_{f_i,p_k} such that $\sum_i N_{f_i,p_k} = N_F$. The probability of any feature f_i in F which is not present in F_{train} is computed from $P(f_i|c_j)$ with $\sum_{k=1}^{k=L} n_{f_i,p_k}$ set to zero.

2.4 Concept Detection

The objective now is to use the learned Bayesian model to identify unlabeled concept instances in the partition tree

[2] This distribution has been shown to perform well in text categorization

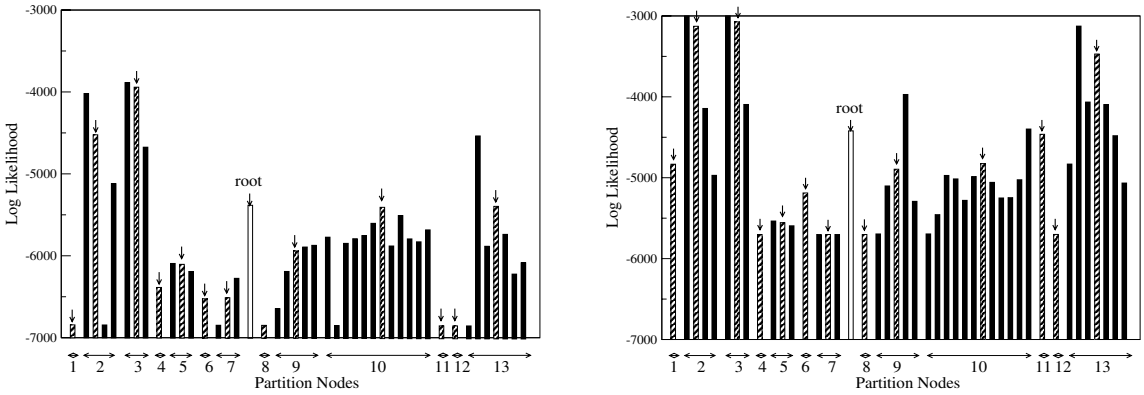

Figure 4. Likelihood values for the first three levels in the Washington Post partition tree for (a) Category News concept, and (b) Taxonomy News concept

of a new HTML document. A straightforward approach is to: (i) compute the likelihood for each concept at every node, (ii) collect all nodes whose likelihood values are greater than a certain threshold, and (iii) select from among them that node with the maximum likelihood value as the concept instance. If there are no nodes with likelihood values above the threshold then the concept does not exist in that document. But this simplistic approach lacks mechanisms to cope with false positives and ambiguities. The latter problem is caused when the same node is the maximum likelihood node for different concepts.

We propose a two-step process to unambiguously label nodes as concept instances. In the first step we generate a set of candidate nodes in the partition tree for every concept. In the second step, we use a novel *bipartite graph* based technique to produce a set of unambiguous $\langle$ concept (c), node (n) $\rangle$ pairs. Each $\langle c, n \rangle$ pair means that the subtree rooted under the node n in the partition tree is an instance of the concept c.

Candidate Generation: Recall that structural analysis aggregates semantically related items under a common node. A consequence of this kind of aggregation is that the semantic content of the subtree rooted at a node is: (i) "similar" to the content in the subtrees rooted at its children, and (ii) "different" from the content in the subtrees of its immediate sibling nodes. We can exploit this property to generate candidate concept instances thus: A node is a candidate concept instance if it's likelihood value is "close" to it's immediate children and "distant" from it's immediate siblings.

As an illustration of this idea let us examine Figure 4(a) and (b). The figure shows the log likelihood values for the category and taxonomy news concepts, respectively, in the partition tree (generated form Washington Post's home page). In both the figures, the unshaded bar in the center represents the root of the partition tree while the checkered bars, with arrows at the top, represent the 13 children of the root, and the shaded bars represent the children of these first level nodes spread equally on either side of the corresponding checkered parent bar. Node 10 corresponds to the category news concept instance while node 3 corresponds to the taxonomy news concept. Observe in Figure 4(a) that the likelihood value for node 10 is more closer to it's children than it's immediate siblings (the checkered bars in 9 and 11). On the other hand observe also that in Figure 4(b) the likelihood value for node 10 is close to both it's children and it's siblings. What this implies is that node 10 is a good candidate for category news concept instance but not for taxonomy news. Also note that node 10 is not the maximum likelihood category news concept node and so a simple maximum likelihood-threshold based method would have failed in this case.

To define the notions "close" and "distant" we use two thresholds: t_{nbr} and t_{child}. We say that a node is "distant" from its neighbours if the mean of the ratio of the deviation of it's likelihood value from each of it's immediate left and right siblings (if they exist) to it's own likelihood value is greater than t_{nbr}. Analogously, we say that the node is "close" to its children if the mean of the ratio of the deviation of it's likelihood value from each of it's children to it's own likelihood value is less than t_{child}.

Based on these two thresholds we can generate the set of candidate nodes for a concept c_i as follows: Let $nbr(p_k)$ denote the set of immediate left and right siblings and $child(p_k)$ denote the set of children of node p_k. Also let $m(p_k)$ denote it's multinomial likelihood value. Then the set of candidate nodes for c_i is:

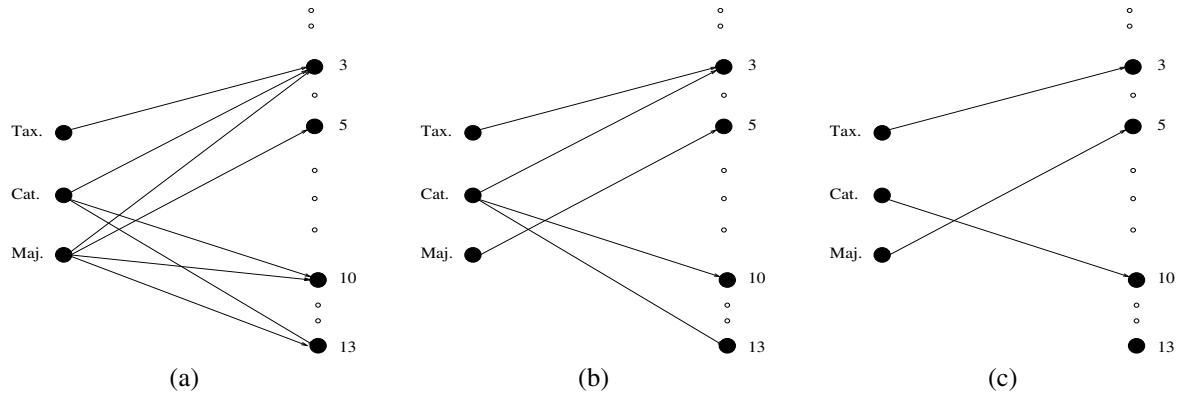

Figure 5. Bipartite Graph between Taxonomy, Category and Major Headlines News and first level nodes in Washington Post partition tree. (a) Original graph, (b) Major Headlines resolved, and (c) Category and Taxonomy resolved

$$Candidate(c_i) = \{p_k | \text{s.t.Avg}_{p_l \in nbr(p_k)} |\frac{m(p_k)-m(p_l)}{m(p_k)}| > t_{nbr}, \text{ and Avg}_{p_c \in child(p_k)} |\frac{m(p_k)-m(p_c)}{m(p_k)}| < t_{chld}\}$$

Ambiguity Resolution: Since the same node can be a candidate for different concepts ambiguities can arise. We can represent the association between concepts and candidate nodes as a bipartite graph – the set of concepts C, and the set of candidate nodes P are the two disjoint sets of vertices in the graph. An edge between $c_i \in C$ to $p_k \in P$ is created if $p_k \in Candidate(c_i)$. Figure 5(a) shows the bipartite graph created by candidate generation for the taxonomy, category, and major headlines news concepts for Washington Post's home page.

The idea behind bipartite graph-based ambiguity resolution is as follows: First we form the set S_i for every concept c_i. S_i consists of nodes that only match c_i. Now pick that node p_k in S_i with the maximum likelihood value to unambiguously represent an instance of the concept c_i. We remove all the other edges from c_i to any $p_l, l \neq k$ from the graph. This computation is repeated until it is not possible to derive any more 1–1 associations between concepts and partition nodes.

Figure 5(a) illustrates the initial bipartite graph between concepts and nodes. Nodes 3, 10, and 13 are matched by different concepts while node 5 only matches the major headlines concept. Consequently, 5 is uniquely associated with major headlines and the edges from major headlines to 3, 10, and 13 are deleted. The residual graph is shown in Figure 5(b). In it, nodes 10 and 13 only match category news. Node 10 is labeled as the instance of category news since its likelihood value is greater than that of node 13. Removing all other edges from category news yields the residual graph in Figure 5(c). A unique association is trivially made between taxonomy news and node 3 and the computation terminates.

It should be noted that the formulation of our ambiguity resolution problem is different from weighted graph bipartite matching algorithms. Techniques for maximal matching on weighted bipartite graphs, for instance the Hungarian Algorithm [23], optimize the sum of the edge weights in the matching. However, we are interested in a maximal *unambiguous* matching which may not correspond to the solution returned by the optimized bipartite matching problem. Our notion of unambiguity places more importance to an edge between a partition node uniquely matched by a concept node even if it's weight, as determined by likelihood values, is low.

3 Experimental Results

We implemented a prototype system based on the techniques described in this paper. For our preliminary experiments we picked: (i)the news domain with (commonly occurring) concepts *major headlines, category*, and *taxonomy news*; (ii) university portals with concepts *university-related news* and *university-related taxonomy*; (iii) travel portal with concepts *hot deals* and *travel-related taxonomy*.

Each Web page was transformed into an unlabeled partition tree via structural analysis. The weighted (structured and unstructured) features were extracted at every node in this tree. For training we picked the home pages of New York Times and CNN for the news domain, the home pages of Columbia and Rutgers University for the universities domain, and the home page of Expedia for the travel domain. The concepts appearing in these example pages were manually identified and accordingly labeled. The t_{nbr} and t_{chld} thresholds were computed by analyzing the likelihood values of the subtrees of the children and siblings of nodes la-

beled as concept instances in the partition trees of the example pages. The trained models were used to detect concept instances in all of the remaining pages in our data set. The results are summarized below.

Table 3 is the recall/precision figures over more than 100 pages from various sites [3]. A ✓ in the P (*Present*) column indicates presence of a concept while − denotes it's absence. A ✓ in the A (*Annotation*) column indicates correct identification while × and − denote incorrect and no identification respectively. All identifications were manually validated. Recall (yield of annotation) is defined as $\frac{\checkmark_A}{\checkmark_P}$ while precision (accuracy of annotation) is defined as $\frac{\checkmark_A}{\checkmark_A + \times_A}$.

The consistent presentation of taxonomic concepts across web sites is reflected in their high recall values. On the other hand the concepts major headlines, university news, and travel deals exhibit, to some degree, varying presentations from site to site and hence suffer from low recall values. Structural features play a dominant role in identifying major headlines and university news concepts. The high recall/precision for these concepts validates the importance of structural features. These results appear to indicate that our ideas on feature extraction, learning statistical models, and concept detection can be seamlessly blended together to identify concept instances with high recall/precision.

In our experiments we measured the impact of using the mix of unstructured and structured features, and the effect of ambiguity resolution on recall and precision. We combined recall and precision metrics into a single measure, namely, the *f-measure* by taking their harmonic mean. The results of these experiments, for all the concepts in the three domains, are summarized in Figure 6(a) and (b) respectively.

Figure 6(a) shows that the mix of structured and unstructured features (shaded bars) is significantly superior to using only words as features (checkered bars). In the travel domain suggestive words like "travel", "hotel", "cars", etc suffice to identify taxonomic instances. Hence words alone as features are adequate. In the news domain quite a few critical words (e.g. "business", "national") appear in both categoric and taxonomic concepts causing ambiguity. So structural features, capturing the different presentation styles of these concepts, become necessary for disambiguation. In Figure 6(b), the checkered bars represent performance when only the the maximum likelihood node is used for identification. Observe the significant improvement with ambiguity resolution especially in the news domain where high degree of ambiguity is present.

[3] results on different home pages from the same site were aggregated in the Table

4 Related Work

The principal areas related to the problem addressed in this paper are the works in partitioning HTML documents into tree-like structures, detecting records in HTML documents by wrapper construction, text categorization techniques, and semantic annotation of Web content.

Heuristics have been proposed to partition HTML documents into tree-like structures so as to facilitate Web browsing on small-screen devices [3, 4], content caching [25], efficient Web search [32], data cleaning [31], and for converting HTML documents into XML data [5, 30]. However, unlike our techniques, none of the above works perform complex structural analysis on the document. Such an analysis is required for grouping semantically related structural entities. These entities are necessary for inducing highly discerning features to detect concept instances in a document with precision.

Extensive research work on constructing wrappers for detecting records in HTML documents exist in the literature [9, 8, 11, 6, 20, 26, 17, 2]. An excellent survey of wrappers appears in [18]. Wrappers are typically learned from syntactic and structural clues of the documents. Consequently, in contrast to semantics-based approaches, they are difficult to scale across documents in a domain.

Learning a concept model from training examples and using this model for detecting instances in documents is closely related to work done on categorization techniques, including Bayesian approaches [21, 19], and topic detection [1]. Excellent surveys on various text classification techniques and their performances appear in [27, 28]. The fundamental difference between the problem of semantic annotation and text categorization is that in the former a single document can contain instances of multiple concepts while categorization assigns a single concept (class) to the entire document. Consequently, unlike any work in text categorization, in the annotation problem we will have to infer the presence of multiple concept instances in a single HTML document. Moreover our work is also concerned with inferring the *logical organization* of a HTML document – the concept hierarchy – which is not addressed in either text classification or topic detection. Another noteworthy point of difference is that text categorization methods do not exploit the (presentational) structure of a document for inducing features (see [29] for a survey on feature selection in text categorization). We do not decouple the content of a document from it's structure and as our experimental results indicate such a coupling is critical for boosting the precision of concept identification.

Recently, a number of research works related to enabling the Semantic Web by enriching Web documents with semantic labels have been reported. In [15, 16, 13, 14] powerful ontology management systems form the backbone

News Portal	Major Headline News		Category News		Taxonomy News	
	P	A	P	A	P	A
New York Times	✓	✓	✓	✓	✓	✓
CNN	✓	✓	✓	✓	✓	✓
Washington Post	✓	✓	✓	✓	✓	✓
Zdnet	✓	×	-	-	✓	✓
Cnet	✓	×	-	-	✓	✓
Citizen Online	✓	-	✓	✓	✓	-
Sun Suntinel	✓	-	✓	✓	✓	-
San Antonio News	✓	-	✓	✓	✓	✓
USA Today	✓	✓	✓	-	✓	✓
ETaiwan News	✓	✓	✓	-	-	-
Financial Times	✓	✓	✓	✓	✓	✓
ABC News	✓	✓	-	-	✓	✓
MSNBC	✓	-	✓	×	✓	✓
Houston Chronicle	✓	✓	✓	✓	✓	×
Chicago Sun Times	✓	-	✓	✓	✓	✓
Yahoo News	-	-	✓	-	✓	✓
Telegraph India	✓	✓	✓	×	✓	-
Independent.co.uk	✓	✓	✓	✓	✓	×
Los Angeles Times	✓	✓	✓	✓	✓	×
Capital Times	✓	✓	✓	✓	✓	✓
Total	19	✓= 12 × = 2	17	✓= 12 × = 2	19	✓= 13 × = 3
Recall (%)	63.16		70.59		68.42	
Precision (%)	85.71		85.71		81.25	

(a)

University	News		Taxonomy	
	P	A	P	A
Columbia	✓	✓	✓	✓
Rutgers	✓	✓	✓	✓
Queens College	✓	✓	✓	✓
Univ of Minnesota	✓	-	✓	-
NCSU	✓	✓	✓	✓
NYU	✓	✓	✓	✓
Southern Methodist	✓	-	✓	✓
Stanford	✓	-	✓	✓
UIUC	✓	-	✓	×
Virginia Polytechnic	✓	✓	✓	✓
Total	10	✓= 6 × = 0	10	✓= 8 × = 1
Recall (%)	60.00		80.00	
Precision (%)	100.00		88.89	

(b)

Travel Site	Deals		Taxonomy	
	P	A	P	A
Expedia	✓	✓	✓	✓
Hotwire	✓	-	✓	✓
Orbitz	✓	✓	✓	✓
Priceline	✓	✓	✓	✓
Yahoo Travel	✓	×	✓	✓
Total	5	✓= 3 × = 1	5	✓= 5 × = 0
Recall (%)	60.00		100.00	
Precision (%)	75.00		100.00	

(c)

Table 1. Experimental results on (a) news portals (b) university home pages (c) travel sites

of tools that support interactive annotation of almost any HTML document. In [7], a system for bootstrapping the semantic web is discussed whereby a large knowledge base is used to create an initial corpus of annotations. In their vision, this initial corpus would spur semantic web application development which in turn would encourage content providers to create more annotations. [12] describes an approach where domain ontologies are used alongwith classifiers and extractors for semantic annotation. While the types and features of metadata extracted in [12] is considerably richer than [7], the system still depends on ontological support for their complete identification. Similarly, [24] describes a linguistic analysis based technique for automatic annotation with respect to a given domain ontology. In contrast to these, our focus is on a *scalable* annotation framework where the model of a semantic concept, learned from training examples, is used to automatically identify it's instances in new documents. Consequently, our framework does not depend on any extensive knowledge base or ontologies to semantically annotate documents.

Finally, unlike any other work, our framework uniquely combines structural analysis, feature extraction and statistical concept detection augmented with ambiguity resolution to assign semantic labels to HTML documents with a high degree of precision and scalability.

5 Discussions

We presented a technique for bootstrapping a process to annotate HTML pages using a set of labeled examples. Creating this set and setting the two thresholds t_{nbr} and t_{chld} are the only two manual steps used in our technique. Even the latter step can be automated by analysis of the log likelihood values computed for the example set. Our technique facilitates a more scalable and automated approach, in contrast to ontology-based methods, to semantic understanding of content-rich Web documents.

Our current notion of structural features is somewhat

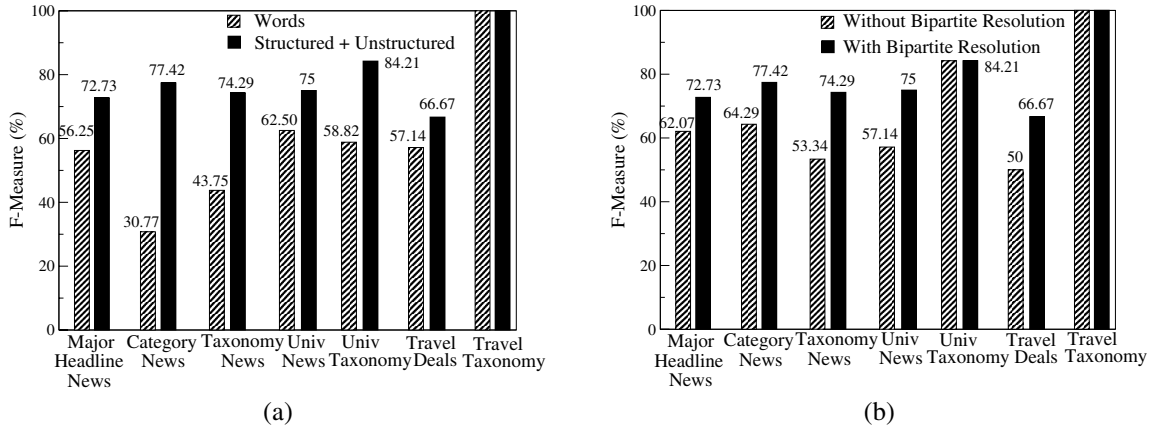

Figure 6. (a) Effect of Feature Extraction on Performance (b) Effect of Bipartite Resolution on Performance

rigid. This may make our technique sensitive to changes in the presentation style of concept instances. Developing more relaxed notions of structural features is a topic for future research. From an experimental perspective, we would like to investigate the effect of the size of the label set on precision of annotation. Currently, we are evaluating the effectiveness of our bootstrapping approach on other domains which are less structured than the ones we have worked with.

Finally it is worthwhile pointing out that using document structure to learn discerning features for concept identification is also relevant to other semi-structured markup languages like XML. Consequently, the methods described in this paper can facilitate semantic integration of XML documents.

References

[1] J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, 2002.

[2] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. *ACM SIGMOD Record*, 26(4), 1997.

[3] O. Buyukkoten, H. Garcia-Molina, and A. Paepcke. Focussed web searching with PDAs. In *Intl. World Wide Web Conf. (WWW)*, 2000.

[4] Y. Chen, W.-Y. Ma, and H.-J. Zhang. Detecting web page structure for adaptive viewing on small form factor devices. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[5] C. Y. Chung, M. Gertz, and N. Sundaresan. Reverse engineering for web data: From visual to semantic structures. In *Intl. Conf. on Data Engineering (ICDE)*, 2002.

[6] W. Cohen, M. Hurst, and L. Jensen. A flexible learning system for wrapping tables and lists in html documents. In *Intl. World Wide Web Conf. (WWW)*, 2002.

[7] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. Tomlin, and J. Yien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[8] D. W. Embley, D. M. Campbell, R. D. Smith, and S. W. Liddle. Ontology-based extraction and structuring of information from data-rich unstructured documents. In *Intl. Conf. on Information and Knowledge Management (CIKM)*, 1998.

[9] D. W. Embley, Y. Jiang, and Y.-K. Ng. Record-boundary discovery in web documents. In *ACM Conf. on Management of Data (SIGMOD)*, 1999.

[10] D. Fensel, S. Decker, M. Erdmann, and R. Studer. Ontobroker: Or how to enable intelligent access to the WWW. In *11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, 1998.

[11] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. M. Breunig, and V. Vassalos. Template-based wrappers in the tsimmis system. In *ACM Conf. on Management of Data (SIGMOD)*, 1997.

[12] B. Hammond, A. Sheth, and K. Kochut. Semantic enhancement engine: A modular document enhancement platform for semantic applications over heterogenous content. In V. Kashyap and L. Shklar, editors, *Real World Semantic Applications*. IOS Press, 2002.

[13] S. Handschuh and S. Staab. Authoring and annotation of web pages in CREAM. In *Intl. World Wide Web Conf. (WWW)*, 2002.

[14] S. Handschuh, S. Staab, and R. Volz. On deep annotation. In *Intl. World Wide Web Conf. (WWW)*, 2003.

[15] J. Heflin, J. A. Hendler, and S. Luke. SHOE: A blueprint for the semantic web. In D. Fensel, J. A. Hendler, H. Lieberman, and W. Wahlster, editors, *Spinning the Semantic Web*, pages 29–63. MIT Press, 2003.

[16] J. Kahan, M. Koivunen, E. Prud'Hommeaux, and R. Swick. Annotea: An open rdf infrastructure for shared web annotations. In *Intl. World Wide Web Conf. (WWW)*, 2001.

[17] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. Wrapper induction for information extraction. In *Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, volume 1, 1997.

[18] A. Laender, B. Ribeiro-Neto, A. da Silva, and J. Teixeira. A brief survey of web data extraction tools. *SIGMOD Record*, 31(2), 2002.

[19] D. Lewis, R. Schapire, J. Callan, and R. Papka. Training algorithms for linear text classifiers. In *ACM Conf. on Informaion Retrieval (SIGIR)*, 1996.

[20] L. Liu, C. Pu, and W. Han. Xwrap: An xml-enabled wrapper construction system for web information sources. In *Intl. Conf. on Data Engineering (ICDE)*, 2000.

[21] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, 1998.

[22] S. Mukherjee, G. Yang, and I. Ramakrishnan. Automatic annotation of content-rich html documents: Structural and semantic analysis. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[23] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1982.

[24] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. Kim - semantic annotation platform. In *Intl. Semantic Web Conf. (ISWC)*, 2003.

[25] L. Ramaswamy, A. Iyengar, L. Liu, and F. Douglis. Automatic detection of fragments in dynamically generated web pages. In *Intl. World Wide Web Conf. (WWW)*, 2004.

[26] A. Sahuguet and F. Azavant. Web Ecology: Recycling HTML pages as XML documents using W4F. In *ACM SIGMOD Workshop on the Web and Databases (WebDB)*, 1999.

[27] F. Sebastiani. Machine learning in automated text categorization. In *ACM Computing Surveys*, 1999.

[28] Y. Yang and X. Liu. A re-examination of text categorization methods. In *ACM Conf. on Informaion Retrieval (SIGIR)*, 1999.

[29] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Intl. Conf. on Machine Learning (ICML)*, 1997.

[30] Y. Yang and H. Zhang. HTML page analysis based on visual cues. In *Intl. Conf. on Document Analysis and Recognition (ICDAR)*, 2001.

[31] L. Yi and B. Liu. Eliminating noisy information in web pages for data mining. In *ACM Conf. on Knowledge Discovery and Data Mining (SIGKDD)*, 2003.

[32] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma. Improving pseudo-relevance feedback in web information retrieval using web page segnmentation. In *Intl. World Wide Web Conf. (WWW)*, 2003.