Learning Structural Semantics for the Web

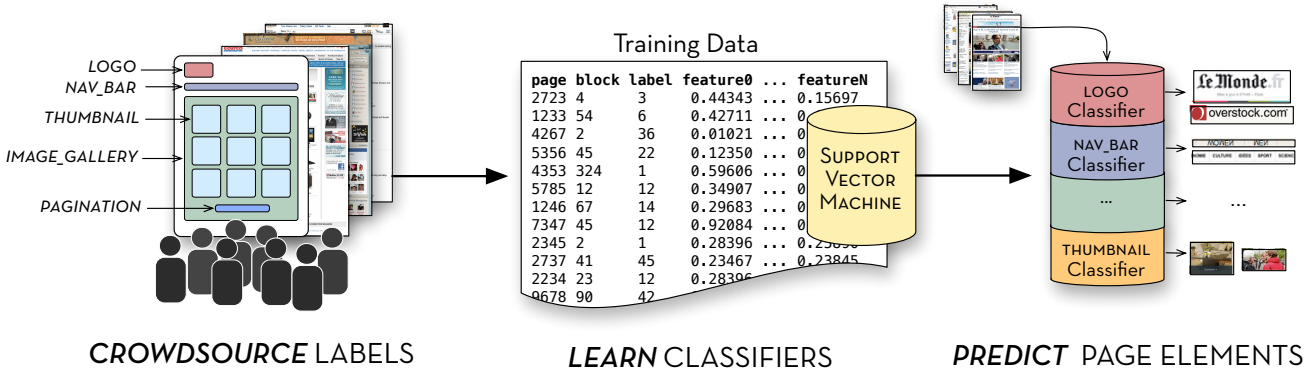

Figure 1. An overview of the pipeline for learning structural semantic classifiers for the Web. First, a set of more than 21,000 labeled page elements are collected from online workers over a corpus of more than 1,400 Web pages. Then, these labels are used to train a set of binary SVM classifiers. Once trained, these classifiers can be used to identify semantic elements in new pages.

ABSTRACT

Researchers have long envisioned a Semantic Web, in which unstructured Web content is replaced by documents that are rich with semantic annotation. Unfortunately, this vision has been hampered by the difficulty of acquiring semantic metadata for Web pages. This paper explores an automatic method for semantifying page elements: using machine learning to train classifiers that can be applied in a post-hoc fashion. We focus on one popular class of semantic identifiers: those concerned with the *structure*—or information architecture—of a page. To determine the set of structural semantics to learn and to collect training data for the learning, we use a crowsourced approach, gathering a large corpus of labeled page elements from a set of online workers. We discuss the results from this collection, and demonstrate that our classifiers provide a promising first step towards learning structural semantics in a general way.

Author Keywords

information architecture, structural semantics, semantic web

ACM Classification Keywords

H.4 Information Systems Applications: Miscellaneous

General Terms

Management, Design

INTRODUCTION

The Web comprises a repository of knowledge on a scale unprecedented in human history. With billions of extant pages, information abounds, but finding, aggregating, and synthesizing information relevant to a particular task remains a difficult and time-consuming problem. One reason for this difficulty is that Web content is largely *unstructured*. Although Web formats provide rich presentation semantics to define the way that machines should display Web data, they typically offer little support for other kinds of automated processing. This dichotomy has engendered a vision of a Semantic Web, in which documents are endowed with sufficient structure to enable machines to "understand" Web content, and respond to complex human requests based on their meaning [2].

Given the difficulties of developing a comprehensive ontology of human knowledge, some recent attempts to semantify to the Web have focused on page *structure* rather than *content*. In HTML 5, the World Wide Web Consortium added semantic tags (*e.g.* <ARTICLE>, <DATE>, <FIGURE>, <SUMMARY>, etc.) to help developers describe the information architecture of pages [15, 9]. These sorts of structural semantics are a small step on the road to a complete semantic "web of data" [16], but they may nonetheless prove invaluable in applications like search [7], retargetting [12], remixing [4], and user interface enhancement [18].

Relying on Web designers to annotate pages. with semantic markup, however, is problematic. Developers—whose primary concern is the display of information —lack compelling incentives to invest time and effort to augment pages with tags that confer no direct presentational benefit. Furthermore, as semantic specifications evolve, pages must be continually reengineered even if their content remains unchanged. An alternative strategy is to allow end-users to add personal semantics

to page data on a case-by-case basis [11, 10], but these highly supervised techniques are difficult to scale to the whole Web.

This paper explores a different tactic for adding structural semantics to Web pages: learning classifiers for page elements from data. With accurate semantic classifiers, pages could be semantified automatically, in a post-hoc fashion, decoupled from the design and authoring process [20]. To this end, we present a classification method based on support vector machines [6], trained on a large collection of human-labeled page elements and employing a feature space comprised of visual and render-time page properties. Since these classifiers make no assumptions about the format, composition, or textual content of Web documents, they can be applied to any HTML page that can be loaded and displayed in a browser.

To identify the set of structural semantics to learn, we take a crowdsourced approach, much like the W3C in selecting semantic tags to add to HTML 5 [13]. Instead of focusing on the way content *producers* view the information architecture of pages, however, we turn our attention to content *consumers* and they way they describe structural semantics. We recruited 400 participants on Amazon's Mechanical Turk, collecting more than 21,000 semantic labels over a corpus of 1200 Web pages. We use these labels to determine the set of classifiers, and to provide training data for the learning.

The paper describes the online label collection study and its results, and demonstrates that SVM-based classifiers provide a promising first step towards learning structural semantics in a general way. We demonstrate how these classifiers can be used to identify semantic elements in existing Web pages, and discuss how post-hoc semantics can improve a variety of Web applications.

ONLINE LABEL COLLECTION

To drive the development of semantic classifiers, we collected a set of labeled page elements in an online study. We recruited 400 US-based workers from Amazon's Mechanical Turk to apply more than 21,000 labels across nearly 1,500 Web pages. Every participant applied semantic labels to at least ten elements on each of five pages. The pages used in the study were drawn from the Webzeitgeist design repository [1], which provides visual segmentations and page features for more than one hundred thousand pages crawled from the Web.

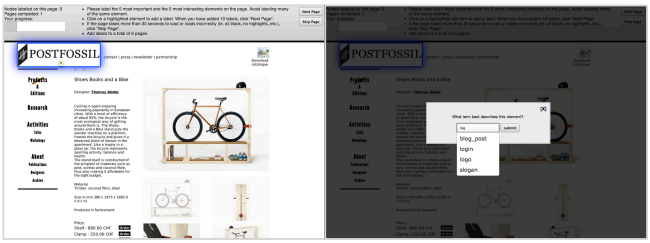

Figure 2. The interface used in the label collection study. Page elements are highlighted in blue upon mouseover (top). After clicking on the highlighted element, users enter semantic labels into a textbox (bottom).

We divided the label collection process into two phases: a focused phase, and a broad phase. In the focused phase, we hand-selected fifty pages from ten popular site genres [8]: e-commerce, news, community, informational, corporate, small company, blog, personal, Web service, and Web resource. A hundred participants each labeled ten of these pages, producing more than five thousand labels and ensuring that many page elements were labeled by more than one person. In the broad phase, three hundred users each labeled five pages chosen randomly from the corpus, producing just over 15,000 labels.

Procedure

Participants were redirected from Mechanical Turk to a tutorial that explained the labeling interface. The instructions directed users to apply semantic labels to the five most *important* and the five most *interesting* elements on the page. Participants were also instructed to avoid labeling many of the same type of element, to encourage diversity in the data set.

Given our focus on structural semantics, workers were told to choose labels that described the *role* of the element in the information architecture of the the page rather than its *content*. For instance, a picture of a silverware set on a shopping page should be labeled PRODUCT_IMAGE instead of SILVERWARE. Workers were also instructed to chose the most specific applicable label, eschewing generalities such a TEXT. To proceed to the labeling task, users were shown a few basic examples of appropriate labels, and required to correctly apply a label to a sample element.

The labeling interface presents workers with a screenshot of a Web page (Figure 6). When a participant hovers the mouse over part of the page, the corresponding element in the page's visual segmentation is highlighted. Clicking on an element allows the user to enter a text label for it, which can be edited later by clicking on the element again. When typing a label, users are prompted with a drop-down list of autocompleted suggestions, sourced from a small pilot labeling study, which they may use or ignore. Workers apply at least ten elements to each page before moving on to the next; after five pages have been labeled, the interface provides an identifier to the worker to verify the tasks' completion.

Results

We collected 21,995 labels across 16,753 distinct elements in 1,490 Web pages. There were 2,657 distinct labels in total, 716 of which occurred more than once, and 629 of which were applied by more than one user. Each participant used 23.6 distinct labels on average ($\min = 3$, $\max = 76$, $\sigma = 9.7$). Excluding labels from the autocomplete list, participants generated an average of nine original label names ($\min = 0$, $\max = 60$, $\sigma = 10$).

The collected labels cover a wide range of structural semantic concepts, with tags as general as IMAGE and as specific as COPYRIGHT. Workers tagged some elements common to all Web pages, such as NAVIGATION, and others that are highly domain specific, such as PRODUCT_IMAGE. The ten most common labels were NAVIGATION_ELEMENT,

NAVIGATION_BAR, LOGO, SEARCH, SOCIAL_MEDIA, ADVERTISEMENT, ARTICLE_TITLE, MAIN_CONTENT, BLOG_POST, and AND CONTACT_LINK, with frequencies ranging from 1,772 to 436. The mean label frequency was 8.3 (min = 1, max = 1772, σ = 65.8).

Figure 3 shows the labels' relative frequencies in a tag cloud. Labels which have direct analogues to any one of the 106 tags in HTML 5 are highlighted in red. The 17 HTML tags to which these labels correspond include <A>, <ADDRESS>, <ARTICLE>, <BLOCKQUOTE>, <BODY>, <CAPTION>, <FIGCAPTION>, <FOOTER>, <FORM>, <H1-H6>, <HEADER>, <HGROUP>, <IMG>, <INPUT>, <NAV>, <TIME>, and <VIDEO>. At a high level, the relatively small overlap between our crowdsourced labels and the set of available HTML tags illustrates the difficulties of developing a semantic ontology that is sufficiently expressive and complete.

LABEL ANALYSIS

To better understand the labels gathered in our study, we performed two different statistical analyses. First, we examined label *co-occurrence*, to determine which labels different workers commonly assign to the same page elements. Second, we examined the *spatial distribution* of labels to determine where certain kinds of page elements commonly appear on a page.

Label Co-occurrence

Since information architecture is far from an exact science, not all people will assign the same semantic label to a given page element. In addition, some workers will use different descriptors to label the same semantic concept. To better understand the way in which labels relate to one another, we created a co-occurrence matrix for the 85 most-frequent labels, each of which was used twenty or more times.

We form an 85×85 symmetric matrix, where the value at (i, j) is the number of times that tag i and tag j were used to label the same page element in our study, normalized by the total number of uses of i and j. Then, the matrix is reordered using Anti-Robinson seriation to form clusters of co-occurring labels along the diagonal [3].

Figure 3. A tag cloud of the 110 most common semantic labels, sized to show relative frequency. The tags highlighted in red have direct analogues in HTML 5.

Figure 4 shows the resulting matrix, with portions of the diagonal magnified to show co-occurring labels. Some of these clusters represent workers using different words to describe the same semantic concept, like COMPANY_LOGO and LOGO (panel A). Other groupings reflect a lack of a clear consensus on the role of particular elements in a page, such as FEATURED_ITEM and PRODUCT_IMAGE (panel I). Some co-occurrences represent the same concept described at different scales, like SITE_TITLE and HEADER (panel C). In general, however, the results validate our study, showing that users are able to label pages in a largely consistent manner.

Spatial Distributions

Another useful way to gain insight about the labels we collected is to examine the spatial distributions of their corresponding page elements. For a given label, we identify the set of page elements to which the label was assigned, and obtain the bounding rectangle for each one from the page's DOM tree. We normalize these rectangles to the range $[0, 1] \times [0, 1]$ to make the coordinates comparable between pages, and rasterize them into a floating-point accumulation buffer. By normalizing the resultant image so that its pixel values sum to one, we obtain a convenient approximation to the two-dimensional spatial probability distribution of the tag. For any given point in the image, the value at that point is the probability of the label appearing in that position on a page.

Figure 5 shows spatial distributions for 28 popular labels. While some distributions are unsurprising (HEADER tags appear almost universally at the top of pages), others give more insight into the structure of Web pages. Note, for instance the strong concentration of LOGIN and SEARCH elements in the upper right corner of pages, the bimodal distribution of ADVERTISEMENT elements between sidebar and header, and the gradual increase of EXTERNAL_LINKS towards the middle of the right sidebar. Taken together, the strong spatial correlations exhibited by many of the collected tags provide strong evidence that learning classifiers for structural semantics may be possible.

LEARNING STRUCTURAL SEMANTIC CLASSIFIERS

To evaluate the feasibility of learning structural semantics from data, we trained binary SVM classifiers for the forty most frequent labels in our study using three different data models. To determine the prediction accuracy of the classifiers, we ran a hold-out test on labeled pages. Finally, we used the learned classifiers to identify and rank semantic elements in a large dataset of pages.

Training

For each distinct label, we constructed a *training* set and a *test* set of page elements. The training set consisted of 80% of the page elements to which the label had been applied (the positive examples), and twice that number of page elements randomly selected from other labels (the negative elements). The test set consisted of the remaining 20% of positively labeled

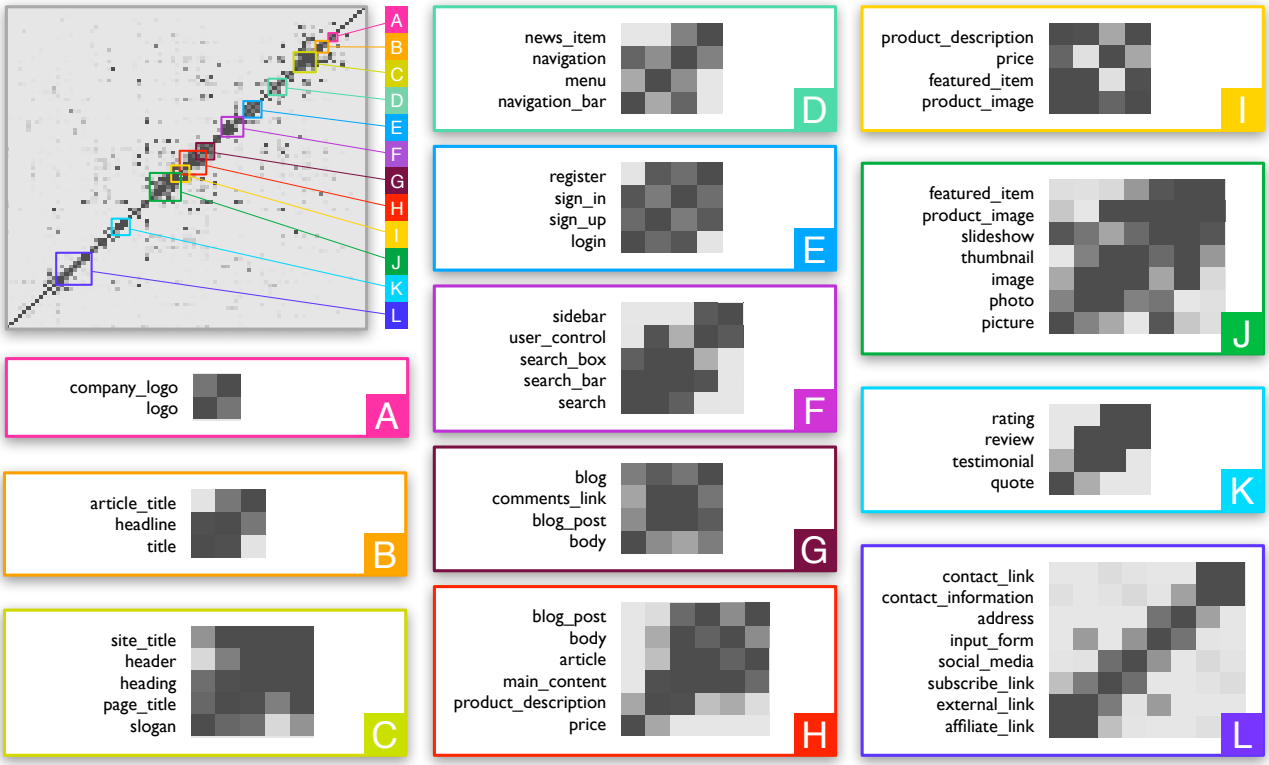

Figure 4. The label co-occurrence matrix, seriated via ARSA [3]. Overlapping sections of the matrix are highlighted and magnified to show labels that frequently occur together.

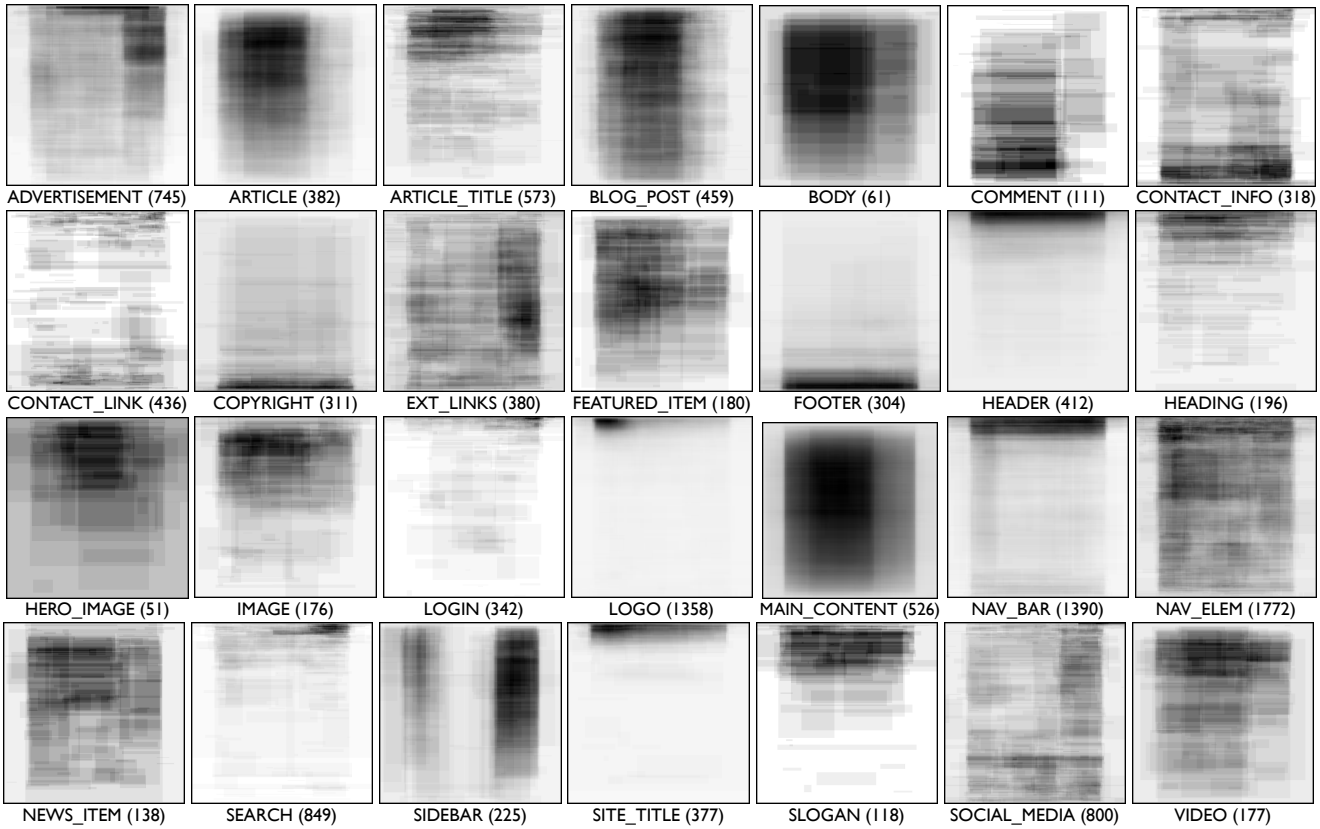

Figure 5. Spatial probability distributions for 28 labels, along with the number of elements used to construct each distribution.

page elements, and twice that number again of randomly selected negative elements.

To drive the learning, each page element was associated with a 1,679-dimensional feature vector provided by the Webzeitgeist repository. These features were drawn from three categories: render-time HTML and CSS properties computed by the DOM ($N = 694$), GIST descriptors computed on element's rendered image (four scales and five orientations per scale on a 4×4 grid; $N = 964$) [14], and simple structural and computer vision properties provided by Webzeitgeist ($N = 22$).

We trained three regularized support vector classification SVMs for each label: one with DOM features, one with GIST features, and a third will all of the features combined. We used LIBSVM to perform the training [5], with radial basis kernels and $\gamma = \frac{1}{1697}$. Once a classifier is trained, it can be used to classify a page element in about $1ms$.

Prediction Accuracy Results

The prediction training and test accuracies for each classifier and data model are shown in the inset table. Test accuracies ranged from 54.9% for COMMENT to 94.7% for ENTIRE_PAGE. The average test accuracy for the DOM, GIST, and ALL models were 74.6%, 71.7%, and 76.6% respectively. The combined model outperformed the DOM- and GIST-alone models for all but nine of the forty labels; examining the training accuracy for those nine shows that this discrepancy is mostly attributable to overfitting. While these results are far from perfect, all of the classifiers do better than random, and most substantially so.

		DOM		GIST		ALL	
Label	#	Train	Test	Train	Test	Train	**Test**
ENTIRE_PAGE	74	91.9	94.7	86.9	75.4	94.6	**94.7**
SEARCH	551	88.7	88.2	82.8	84.5	91.8	**91.5**
FOOTER	186	83.0	78.0	74.6	75.9	90.0	**89.4**
IMAGE	169	84.0	81.0	79.7	79.4	86.4	**88.9**
SIDEBAR	133	86.0	84.8	82.7	84.8	86.7	**87.9**
COPYRIGHT	206	86.1	82.1	75.7	76.3	88.4	**87.8**
NAVIGATION_BAR	901	80.7	83.4	74.6	72.7	86.5	**87.4**
LOGO	770	80.7	84.0	77.9	77.6	87.0	**87.3**
ARTICLE_TITLE	373	84.2	82.8	80.3	82.4	86.7	**87.1**
MAIN_CONTENT	350	82.8	82.8	78.9	80.1	83.0	**83.1**
PRODUCT_IMAGE	65	79.5	79.2	77.4	75.0	85.1	**81.3**
THUMBNAIL	83	84.7	76.2	79.5	73.0	86.7	**81.0**
HEADING	134	79.4	87.3	74.9	67.6	77.9	**80.4**
ARTICLE	237	75.2	72.9	81.4	79.7	85.9	**80.2**
LOGIN	222	80.8	73.9	77.8	74.5	84.5	**78.8**
ADVERTISEMENT	487	79.2	76.2	75.2	72.4	85.6	**77.9**
NAV_ELEMENT	1138	75.9	75.4	72.6	73.5	78.6	**77.8**
VIDEO	107	74.8	71.6	81.9	72.8	81.9	**77.8**
BLOG_POST	259	75.5	72.3	73.6	73.8	81.9	**77.4**
HEADER	265	78.5	77.3	69.2	67.2	80.8	**77.3**
CONTACT_LINK	278	76.7	75.2	74.8	71.0	80.7	**76.2**
SOCIAL_MEDIA	514	75.0	72.7	76.5	70.1	82.2	**75.0**
SITE_TITLE	272	75.9	74.0	76.1	70.6	81.3	**75.0**
DATE	91	79.1	69.6	78.0	73.9	79.5	**73.9**
IMAGE_GALLERY	94	76.6	76.8	75.2	71.0	82.3	**73.9**
RECOMM_LINKS	137	69.8	66.7	68.9	66.7	76.4	**73.5**
CONTACT_INFO	183	77.2	67.4	69.9	68.1	79.6	**73.2**
LANG_SELECT	70	77.1	68.6	75.2	70.6	83.3	**72.5**
PROD_DESC	232	77.2	71.8	74.7	68.4	77.0	**72.4**
SLOGAN	79	69.6	63.3	70.5	70.0	76.4	**70.0**
AUTHOR	94	72.7	62.3	67.4	68.1	70.6	**69.6**
SUBSCRIBE_LINK	158	68.1	67.5	69.6	67.5	71.3	**68.4**
FEATURED_ITEM	107	71.3	71.6	73.5	58.0	76.9	**66.7**
COMMENTS_LINK	93	80.6	71.0	67.7	66.7	70.3	**66.7**
AFFILIATE_LINK	78	66.7	66.7	66.7	66.7	66.7	**66.7**
EXTERNAL_LINKS	270	67.2	66.2	68.9	66.2	73.8	**66.2**
SIGN_UP	134	74.9	65.7	69.2	66.7	73.9	**65.7**
NEWS_ITEM	121	68.9	65.6	71.1	64.4	72.2	**65.6**
DOWNLOAD_LINK	77	73.6	63.2	66.7	66.7	74.5	**64.9**
COMMENT	70	77.6	76.5	75.7	56.9	78.6	**54.9**

Identifying Structural Elements

To show the learned classifiers in action, we ran twelve of them across a database of $500,000$ page elements spanning $3,000$ pages, and ranked the results in order of decreasing probability [21]. A few representative results for each classifier are shown in 6; page elements that appeared to be misclassified are marked with ⊗.

These examples allow us to gain some insight into the performance of the classifiers. Most of the highly-ranked elements are classified correctly, despite their diverse contexts and compositions. Given that our classifiers are trained only on visual and structural data, their expressive power provides support for the notion that structural semantics can be learned without requiring more complex content-based semantics (see, for instance, SLOGAN). Many of the errant classifications are subtle, and might plausibly confuse a human worker: see for instance ARTICLE_TITLE, which classifies several titles that are not, strictly speaking, associated with articles; and NAVIGATION_BAR, which identifies page elements filled with links directing users to *other* sites.

DISCUSSION AND FUTURE WORK

We have presented a promising first step towards adding structural semantics to the Web in a post-hoc manner, demonstrating that a relatively simple machine learning technique can be used to identify semantic elements in pages when trained on a large corpus of human annotations. There remain, however, several avenues for future work.

In the first place, it is important to note that our classifiers cannot realistically be used to enable the one-click annotation of pages in their current form. With an average of 1380 DOM nodes per page, even a 99.9% per-classifier error rate would yield several misclassified nodes on every page: an unacceptable result for many real-world applications (Figure 7).

Several possibilities for improving the learning come to mind. Using our classifiers to bootstrap an online learning process is one obvious approach, likely to significantly reduce overfitting and greatly simplify the acquisition of additional training data. Adding more sophisticated structural and computer vision features is another: estimates of foreground and background area, for instance, might prove useful in recognizing logos, while structural features like "number of links to external domains" could improve the classification of navigation bars.

Another promising approach is to turn to machine learning methods that make better use of *structure*. In our current for-

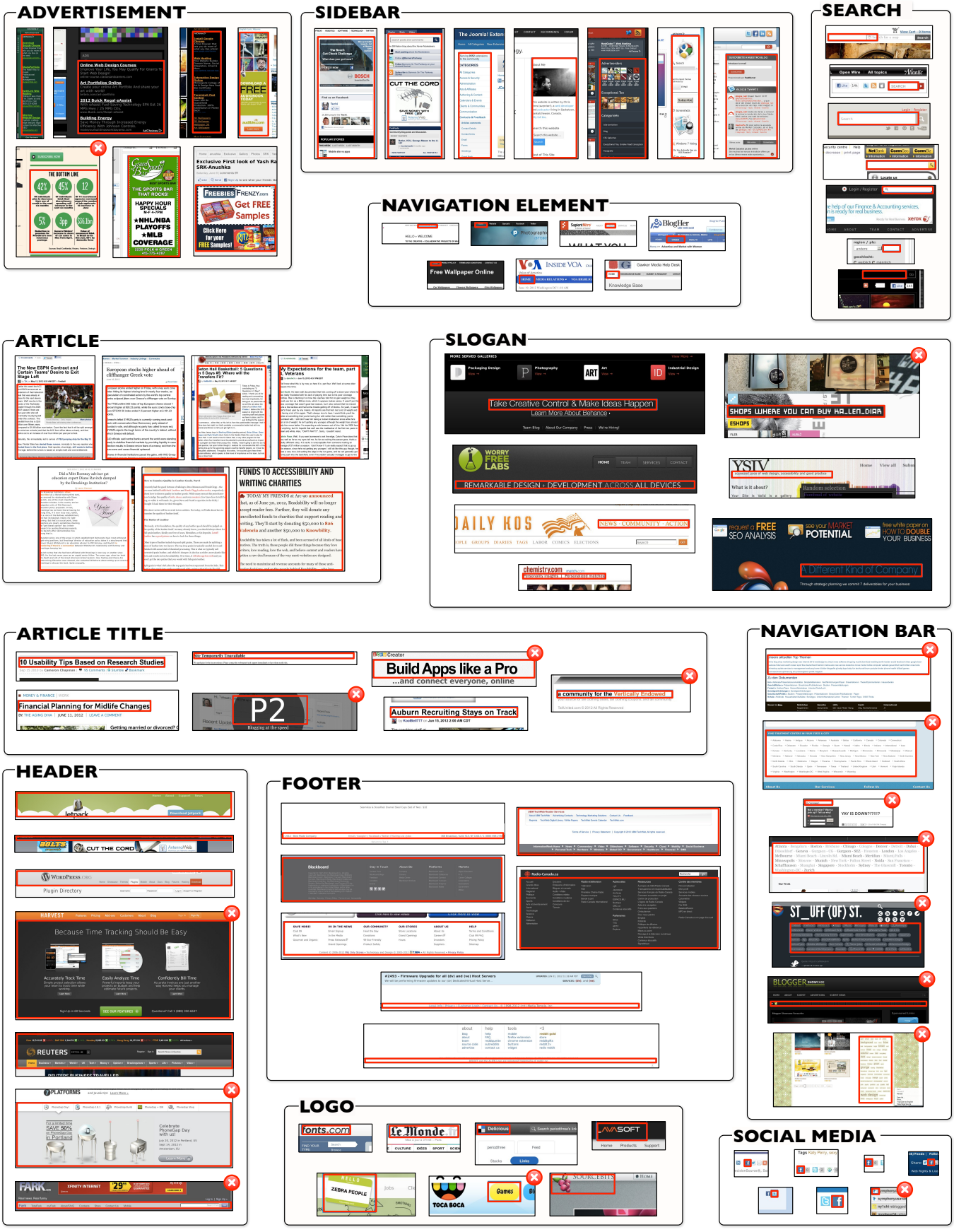

Figure 6. Shown are the top 7 out of 0.5 million nodes ranked by the classifier for each of 12 labels. The nodes were ranked based on on the predicted probability estimate, and a maximum of one node per distinct page is displayed.

Figure 7. Eight of our learned classifiers used to identify structural semantic elements on a novel page with 67 DOM elements. Correct classifications are shown in blue, incorrect in red, and missed elements are in dashed red.

malism, classification proceeds independently on each page node, an assumption that is obviously faulty. Structured SVMs could be used to predict labels holistically, to the entire page as a whole [19]. In addition, deep learning techniques, like those based on recursive neural networks, might allow the development of a more structurally sensitive feature space, making it easier to classify elements whose semantic function is highly dependent on its relation to other elements in the page hierarchy [17].

References

[1] ANONYMOUS. Webzeitgeist: design mining the web. In: *submission*. 2012.

[2] BERNERS-LEE, Tim, HENDLER, James, and LASSILA, Ora. The semantic web. *Scientific American* (May 2001), 35–43.

[3] BRUSCO, Michael J and STAHL, Stephanie. An algorithm for extracting maximum cardinality subsets with perfect dominance or anti-Robinson structures. *British journal of mathematical and statistical psychology* 60, 2 (2007), 377–393.

[4] CHAN, Bryan, WU, Leslie, TALBOT, Justin, CAMMARANO, Mike, and HANRAHAN, Pat. Vispedia: interactive visual exploration of Wikipedia data via search-based integration. *Proc. InfoVis* 14 (6 2008), 1213–1220.

[5] CHANG, Chih-Chung and LIN, Chih-Jen. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2 (3 2011), 27:1–27:27.

[6] CORTES, Corinna and VAPNIK, Vladimir. Support-Vector Networks. In: *Machine Learning*. 1995, 273–297.

[7] DONTCHEVA, Mira, DRUCKER, Steven M., SALESIN, David, and COHEN, Michael F. Relations, cards, and search templates: user-guided web data integration and layout. In: *Proc. UIST*. ACM, 2007, 61–70.

[8] DUYNE, Douglas K. van, LANDAY, James A., and HONG, Jason I. *Design of Sites*. 2nd ed. 2009.

[9] HICKSON, Ian. URL: **http://dev.w3.org/html5/spec-author-view/**.

[10] HOGUE, Andrew and KARGER, David. Thresher: automating the unwrapping of semantic content from the World Wide Web. In: *Proc. WWW*. ACM, 2005, 86–95.

[11] HUYNH, David, MAZZOCCHI, Stefano, and KARGER, David. Piggy bank: experience the semantic web inside your Web browser. In: *Proc. ISWC*. 2005, 413–430.

[12] KUMAR, Ranjitha, TALTON, Jerry O., AHMAD, Salman, and KLEMMER, Scott R. Bricolage: example-based retargeting for web design. In: *Proc. CHI*. ACM, 2011.

[13] MONROE, Lee. *How HTML5 element names were decided*. Apr. 2011. URL: **http://www.leemunroe.com/html5-element-names**.

[14] OLIVA, Aude and TORRALBA, Antonio. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision* 42, 3 (2001), 145–175.

[15] ROSENFELD, Louis and MORVILLE, Peter. *Information architecture for the world wide web*. 2nd ed. O'Reilly & Associates, Inc., 2002.

[16] SHADBOLT, N., BERNERS-LEE, T., and HALL, W. The semantic web revisited. *IEEE Intelligent Systems* 21, 3 (2006), 96–101.

[17] SOCHER, Richard, MANNING, Christopher, and NG, Andrew. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks. In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 2010.

[18] TOOMIM, Michael, DRUCKER, Steven M., DONTCHEVA, Mira, RAHIMI, Ali, THOMSON, Blake, and LANDAY, James A. Attaching UI enhancements to websites with end users. In: *Proc. CHI*. ACM, 2009.

[19] TSOCHANTARIDIS, Ioannis, JOACHIMS, Thorsten, HOFMANN, Thomas, and ALTUN, Yasemin. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 6 (2005), 1453–1484.

[20] WU, Fei and WELD, Daniel S. Autonomously semantifying wikipedia. In: *Proc. CIKM*. ACM, 2007, 41–50.

[21] WU, Ting fan, LIN, Chih-Jen, and WENG, Ruby C. Probability Estimates for Multi-class Classification by Pairwise Coupling. *Journal of Machine Learning Research* 5 (2003), 975–1005.