ISSN 2047-3338

A Learning Nested Block Approach to Web Pages

U. Vinod Kumar[1], Shaik Yacoob[2], Sudam Sekhar Panda[3], Ch. Rajaramesh[4]
[1,2,3,4]Regency Institute of Technology, Yanam
{vinod.uppalapu, shaikyacoob}@gmail.com, {shekhar_regency, ch_rajaramesh}@yahoo.com

***Abstract*– Content of the web page is the textual and graphical information that related to the topic of the page, which is the focus of web data mining and information retrieval. For web pages, the page content is the target of word-segmentation and indexing for search engine, corpus collection of news, reviews, blogs, etc. for knowledge management researches. Extracting content of the web pages correctly and efficiently improves the accuracy of following analysis for it significantly reduces the noise in the pages, and also alleviates the workload of indexing and segmentation. In this works, no uniform approach or model is presented to measure the importance of different nested portions in web pages. Through a user study, we found that people do have a consistent view about the importance of blocks in web pages. In this paper, we investigate how to find a model to automatically assign importance values to nested blocks in a web page. We define the block importance estimation as a learning problem. First, we use the VIPS (Vision-based Page Segmentation) algorithm to partition a web page and block in the webpage into semantic blocks with a hierarchy structure. Then spatial features (such as position, size) and content features (such as the number of images and links) are extracted to construct a feature vector for each block and nested blocks. . Based on analyzing the features of the pages, this approach could effectively extract contents from web pages. Experiments show good results comparing to related works.**

***Index Terms*– Block Importance Model, page Segmentation, Web Mining and Classification**

I. INTRODUCTION

THE Web provides people a convenient media to disseminate all kinds of information. With the rapid increase of information spreading on the Web, an effective method for users to discern the useful information from the junk is in great demand. There is a need to differentiate good pages that are more authoritative from sporadic ones. Within a single web page, it is also important to distinguish valuable information from noisy content that may mislead users' attention. The former has been well studied by link analysis techniques such as Block Importance [1] and Page Rank [2]. However, up to date, there is no effective technique for the latter aspect. Most applications consider the whole web page divided in to Blocks and a given different importance to the blocks but the content in the Block is treated equally.

Obviously, the information in a web page is not equally important and information inside the Block is also not equally important. For example the headline in a news web site is much more attractive to users than the navigation bar. And users hardly pay attention to the advertisement or the copyright when they browse a web page. Therefore, different information inside a web page has different importance weight according to its location, occupied area, content, etc. Thus, it is of great advantage to have a technique which could automatically analyze the information in a web page and assign importance measures for different regions in the web page.

To assign importance to a region and sub region in a web page, we first need to segment a web page into a set of blocks and again divide that block into blocks (nested blocks). There are several kinds of methods for web page segmentation. Most popular ones are DOM-based segmentation [4], location-based segmentation [9] and Vision-based Page Segmentation (VIPS) [17][3]. These methods distinguish each other by considering various factors as the partition basis. Though these methods take one step ahead to look down into the structure of a web page instead of treating it as a unit, they do not differentiate the importance of the blocks in a page and still treat them uniformly.

To solve this problem, we propose a nested block importance model in this paper to assign importance values to different blocks inside the blocks in a page. First, the Vision-based Page Segmentation (VIPS) algorithm is used to segment a page into blocks according to the content coherence by analyzing the visual layout of the page. Then, block features (including spatial features and content features) are extracted to represent the blocks. The same process is applied for block inside another bock. Finally, based on these features, we use SVM and neural network methods to learn general block importance models.

The main contributions of this paper are:

i). A comprehensive user study is conducted to validate that people do have consistent opinions on the importance of different regions and sub regions in web pages.

ii). A block importance model is proposed to automatically assign importance weights to different regions and sub regions in a web page. This model takes into account both spatial features and content features.

iii). Two methods based on neural network and SVM for importance assignment are proposed.

iv). Many promising applications of the nested block importance model are discussed.

The rest of the paper is organized as follows. In Section II, previous related works are described. In Section III, we introduce the page segmentation methods, block segmentation especially the VIPS method. Section IV is the user study we conducted to validate that people do have consistent opinions about the importance of different regions and sub regions in web pages. In Section V, we introduce the details of the nested block importance model, including the features and learning methods we use. Experimental evaluation is provided in Section VI to assess the performance of our model and some insights are also given here. Finally, we discuss conclusion how the applications of our work in Section VII and make a conclusion in Section VIII.

II. PREVIOUS WORK

Previous related works about judging the importance of different parts in a web page can be classified into two classes. One class of techniques aims to detect the patterns among a number of web pages from the same web site. The common idea of these works is that "in a given web site, noisy blocks usually share some common contents and presentation styles" [15]. Bar- Yossef et al. define the common parts among web pages as template [1]. When web pages are partitioned into some "pagelets" based on some rules, the problem of template detection is transformed to identify duplicated "pagelets" and count frequency. Their experimental results show that template elimination improves the precision of the search engine Clever at all levels of recall. Another content-based approach is proposed by Lin and Ho [10]. Their system, Info Discover, partitions a web page into several content blocks according <TABLE> tags. Terms are extracted as features and entropy is calculated for each term and block entropy is calculated accordingly. An entropy-threshold is selected to decide whether a block is informative or redundant. Different from these two works, Yi and Liu make use of the common presentation style [15], [16].

A Style Tree is defined to represent both layout and content of a web page. Node importance is defined as the entropy of the node in the whole Style Tree for a site. By mapping a page of this site to the Site Style Tree, noisy information in the page is detected and cleaned. Their experimental results show that the noise elimination technique is able to improve data mining tasks such as clustering and classification significantly. The other class of techniques tries to detect important regions in a single web page. Gupta et al. [8] have proposed a DOM-based content extraction method to facilitate information access over constrained devices like PDAs. They implemented an advertisement remover by maintaining a list of advertiser hosts, and a link list remover based on the ratio of the number of links and non-linked words. But this rule-based method is relatively simple. For a portal website like www.msn.com which is full of links, the rule would remove almost all useful contents. Besides of purely utilizing contents, Kovacevic et al. [9] used visual information to build up a M-Tree, and further defined heuristics to recognize common page areas such as header, left and right menu, footer and center of a page. In [4], a function model called FOM is used to represent the relationships between features and functions. This work is close to our works, but it is rule-based and cannot deal with dozens of features with complicate correlations.

III. PAGE SEGMENTATION

Several methods have been explored to segment a web page into regions or blocks [4][10]. In the DOM-based segmentation approach, an HTML document is represented as a DOM tree. Useful tags that may represent a block in a page include P (for paragraph), TABLE (for table), UL (for list), H1~H6 (for heading), etc. DOM in general provides a useful structure for a web page. But tags such as TABLE and P are used not only for content organization, but also for layout presentation. In many cases, DOM tends to reveal presentation structure other than content structure, and is often not accurate enough to discriminate different semantic blocks in a web page. Another intuitive way of page segmentation is based on the layout of webpage.

In this way, a web page is generally separated into 5 regions: top, down, left, right and center [9]. The drawback of this method is that such a kind of layout template can not be fit into all web pages. Furthermore, the segmentation is too rough to exhibit semantic coherence. Compared with the above segmentation, Vision-based page segmentation (VIPS) excels in both an appropriate partition granularity and coherent semantic aggregation. By detecting useful visual cues based on DOM structure, a tree-like vision based content structure of a web page is obtained.

The granularity is controlled by a degree of coherence (DOC) which indicates how coherence each block is. VIPS can efficiently keep related content together while separating semantically different blocks from each other. Visual cues such as font, color and size, are used to detect blocks. Each block in VIPS is represented as a node in a tree. The root is the whole page; inner nodes are the top level coarser blocks; children nodes are obtained by partitioning the parent node into finer blocks; and all leaf nodes consist of a flat segmentation of a web page with an appropriate coherent degree. The stopping of the VIPS algorithm is controlled by a predefined DOC (PDOC), which plays a role as a threshold to indicate the finest granularity that we are satisfied. The segmentation only stops when the DOCs of all blocks are no smaller than the PDOC. Fig. 2 shows the result of using VIPS to segment a sample CNN web page.

A. Block Segmentation

The Block Segmentation is process of applying the page segmentation process repeatedly for all blocks in the webpage. This is to give a importance to sub blocks in the webpage when the page contains more information than the simple or normal webpage contains.

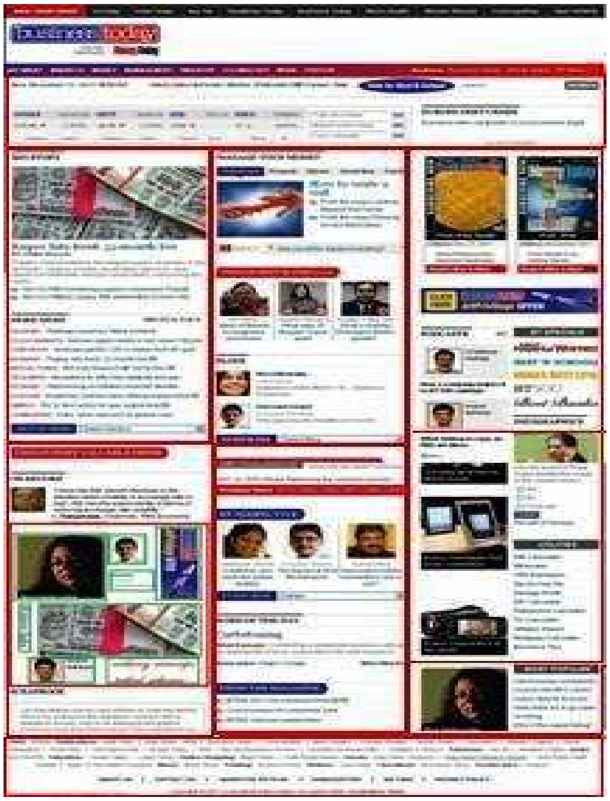

Fig. 1. VIPS segmentation of a part of sample web page

IV. A USER STUDY OF BLOCK MPORTANCE

Since our task is to learn an importance model for web pages, a critical question will be raised first: do people have consistent opinions about the importance of the same block in a page?

Importance is a concept different from attention. Attention is a neurobiological conception. It means the concentration of mental powers on an object, a close or careful observing or listening [11]. At the first sight of a web page, attention may be caught by an image with bright color or animations in advertisement, but generally such an object is not the important part of the page. Also, attention is quite subjective if considering users' purposes and preferences. For example, one user may go to a portal website to see the news headline, and another user may first check the stock quotes in the same page. It is difficulty to find a general model describing such subjective importance definition.

Here, our target is to define block importance from an objective point of view. Block importance should reflect the correlation degree between a block and the theme of the web page. Since the theme is determined by the web page's authors, an objective importance definition is actually based on the author's view but the user's views.

The research in [1] has shown that 3/5 people have the same opinion on how the important of the blocks are. In the similar way we conduct a user study to testify whether such an objective importance model exists. The tool used in the user study is illustrated in Fig. 3. First, a web page is segmented into a nested block structure by using the VIPS algorithm. For each page, the VIPS process is stopped at the point when further segmentation will destroy the semantic integration of blocks. Then all of the leaves blocks form a partition of the page.

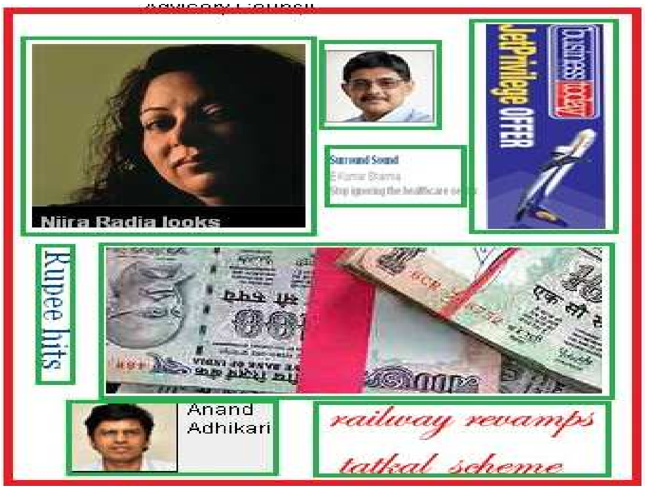

Fig. 2. The block importance labeling tool for conducting the user study

The research in [1] has shown that 3/5 people have the same opinion on how the impotents of the blocks are. In the similar way we conducted has shown that 600 web pages from 405 sites in 3 categories in yahoo: news, science and shopping. Each category includes 200 pages. Treated homepage and inner pages of a website as different pages, thus the impact of websites is ignored here. After segmenting these 600 pages, totally got 4539 blocks

Then 5 human assessors to manually label each block with the following 4-level importance values:

Level 1: Noisy information such as advertisement, copyright, decoration, etc.

Level 2: Useful information, but not very relevant to the topic of the page, such as navigation, directory, etc.

Level 3: Relevant information to the theme of the page, but not with prominent importance, such as related topics, topic index, etc.

Level 4: The most prominent part of the page, such as headlines, main content, etc.

When importance is divided into 4 levels, for 92.9% blocks, a majority of assessors (3/5) have the same opinion on how important these blocks are. When level 2 and 3 are merged as one level, the assessors achieved majority agreement for 99.5% majority agreement for all of the blocks (Table 1).

In Table 2 and Table 3, we list the evaluation results when combining any two importance levels or three importance levels into one single level. In these evaluations, we intend to check which levels are difficult and which are easy for users to differentiate. Table 2 shows that when level 1 and 2 are merged, the highest percentage of 4/5 agreement and 5/5 agreement can be obtained, and when level 2 and 3 are merged, the highest 3/5 agreement is reached. These phenomena indicate that levels (1, 2) and (2, 3) are relatively confused for the assessors while level 4 can be most clearly identified. Accordingly, when levels 1, 2, 3 are merged together, the assessors reached very high agreement (Table 3). Also, it is interesting to notice that when (1, 4) and (2, 3) are merged to 2 levels, the consistency also is quite good. The

Table 1: Agreement on 4-level, 3-level and 2-level importance

Levels	**3/7 agreement**	**4/7 agreement**	**5/7 agreement**	**6/7 agreement**	**7/7 agreement**
1,2,3,4	0.920	0.540	0.268	0.089	0.001
1,(2,3),4	0.989	0.699	0.389	0.187	0.057
(1,2,3),4	0.999	0.898	0.810	0.701	0.600

Table 2: Agreement on all kinds of 3-levelimportance

Levels	**3/7 agreement**	**4/7 agreement**	**5/7 agreement**	**6/7 agreement**	**7/7 agreement**
(1,2),3,4	0.952	0.77	0.568	0.356	0.102
1,(2,3),4	0.983	0.735	0.419	0.221	0.012
1,2,(3,4)	0.960	0.619	0.322	0.121	0.005
(1,3),2,4	.0.955	0.558	0.251	0.101	0.004
1,3,(2,4)	0.954	0.565	0.253	0.098	0.014
(1,4),2,3	0.931	0.549	0.26	0.09	0.001

Table 2: Agreement on all kinds of 2-levelimportance

Levels	**3/7 agreement**	**4/7 agreement**	**5/7 agreement**	**6/7 agreement**	**7/7 agreement**
(123),4	1	0.940	0.840	0.754	0.629
1,(2,3,4)	1.1	0.858	0.592	0.312	0.155
(1,3,4),2	1.2	0.689	0.356	0.101	0.009
(1,2,4),3	1.01	0.801	0.598	0.321	0.115
(1,2),(3,4)	1	0.746	0.49	0.26	0.11
(1,4),(2,3)	1	0.858	0.648	0.412	0.201
(1,3),(2,4)	1	0.599	0.269	0.001	0.012

reasons may lie in that most important blocks and most unimportant blocks can be more easily distinguished from those are between, and levels 2 and 3 are the most blurry zones to be distinguished. So, in practice, we usually combine levels 2 and 3.

The user study clearly demonstrates that users do have consistent opinions when evaluating the importance of blocks, and it is meaningful to explore a way to model the importance of web page blocks.

V. NESTED BLOCK IMPORTANCE MODEL

Web page designers tend to organize their content in a reasonable way: giving rominence to important things and deemphasizing the unimportant parts with proper features such as position, size, color, word, image, link, etc. A block importance model is a function to map from features to importance for each block, and can be formalized as:

<Block features> ->block importance->nested block importance

A. Nested Block Features

Let us take a look at the web page in Fig. 1 again. What features are used to differentiate the important parts from unimportant parts? Typically, web designer would like to put the most important information in the center, put the navigation baron the header or the left side and the copyright on the footer (the information in the solid circles is ore importance than those in the dashed circle in Fig. 1). Thus, the importance of a block can be reflected by spatial features like position, size, etc. On the other hand, the contents in a block are also useful to judge block importance. For example, the spatial features of both of the two solid circles in Fig. 1 are similar. But one contains a picture, a highlighted title and some words to describe a news headline and another contains pure hyperlinks pointing to other top stories. Based on the contents of the blocks, it is possible to differentiate their importance. Therefore, we also include content features in Nested.

1) Spatial Features

With the segmentation of VIPS, each block is described by a rectangle located in the page. Spatial features of a block are made up of four features:

{Block Center X Block Center Y, Block Rect Width, Block Rect Height}.

Block Center X and *Block Center Y* are the coordinates of the center point of the block and *Block Rect Width, Block Rect Height* are the width and height of the block.

Such spatial features are called *absolute spatial features* since they directly use the absolute values of the four features. But using absolute values may make it hard to compare the features from different web pages. For example, a big block in a small page will always be taken as small block when comparing it with the blocks in a big page. So, by using the width and height of the whole page to normalize the absolute features, we transform theminto *relative spatial features*, as given below:

{Block Center X/ Page Width, Block Center Y/Page Height,

BlockRectWidth/PageWidth, BlockRectHeight/PageHeight}.

We found that size normalization brings up another problem. For some long pages with height times larger than the screen height (e.g., the page in Fig. 1 or pages longer than it), after normalization, some important blocks on the top part (i.e., blocks displayed in the first screen, such as the blocks in the solid circles in Fig. 1) may be transformed into blocks located at the top of the page with quite small height. In these cases, the spatial features of these important blocks are very similar to the spatial features of the unimportant blocks such as advertisements in short pages. The point here is that, for a long page, the content in the first screen is most important and we should avoid normalizing them with the height of the whole page. Width normalization does not have the same problem since few pages have widths bigger than the screen. Based on the above observations, we further modified the relative spatial features into *window spatial features*. Instead of using the height of the whole for normalization, we use a fixed-height window instead.

BlockRectHeight= BlockRectHeight / WindowHeight;

Also, feature BlockCenterY is modified as:

$$BlockCenterY = \begin{cases} BlockCenterY/(2*HeaderHeight); \\ \qquad if \quad BlockCenterY < HeaderHeight \\ 0.5; \qquad if \quad HeaderHeight < BlockCenter \\ \qquad\qquad Y < PageHeight - FooterHeight \\ 0.5 + (PageHeight - BlockCenterY)/(2*FooterHeight); \\ \qquad otherwise \end{cases}$$

Where *HeaderHeight* and *FooterHeight* are predefined constant values about the heights of header and footer of a page.

2) Content Features

The following 9 features are used to represent the content of a block:
{Img Num, Img Size, LinkNum, Link Text Length, Inner TextLength, Interaction Num, InteractionSize, FormNum, FormSize}
ImgNum and *ImgSize* are the number and size of images contained in the block. *LinkNu* and *LinkTextLength* are the number of hyperlinks and anchor text length of the block. *InnerTextLength* is the number of words. *InteractionNum,* and *InteractionSize* are the number and size of elements with the tags of <INPUT> and <SELECT>. *FormNum* and *FormSize* are the number and size of element with the tag <FORM>. Like spatial features, all of these features are supposed to be related to the importance. For example, an advertisement may contain only images but texts, and a navigation bar may contain quite a few hyperlinks. These content features are also normalized by the feature values of the whole page. For example, the *Link Num* of a block is normalized by the link number of the whole page.

B. Learning Nested Block Importance

Basically, there are two possible ways to deduce block importance from block features. First, we can design some empirical rules to infer the block importance from its features, such as size, position, etc. There are also some works addressing the problem of block function identification. In [4], an automatic rule-based approach is presented to detect the functional property and category of object. However, this method is unstable and it is very difficult to manually compose rules in functions of dozens of features. Therefore, in this paper, we adopt the second way, learning from example. Specially, some blocks are pre-labeled by several people and thus each labeled block can be represented as (**x**, y) where **x** is the feature representation of the block and y is its importance (label). The set of labeled blocks usually refers to training set *T*.
Thus, problem becomes to find a function *f* such that

$$\sum_{(\mathbf{x},y)\in \mathrm{T}} \left| f(\mathbf{x}) - y \right|^2$$

is minimized. Note that, if *y* is discrete then this is a classification problem and it becomes a regression problem if *y* is continuous. There are various existing learning methods. In our work, we use two learning methods to build the block importance model. One is the neural network learning method when we treat it as a regression problem. Another is the SVM learning method when viewing it as a classification problem.

1) Regression by Neural Network

When the labels are continuous real number, neural network learning can be applied for learning the optimal *f** which is given by minimizing the following cost function:

$$f^* = \arg\min_f \sum_{i=1}^{m} \left\| f(\mathbf{x}_i) - y_i \right\|^2$$

where *m* is the number of blocks in training dataset. Clearly, this is a multivariate nonparametric regression problem, since there is no *a priori* knowledge about the form of the true regression function which is being estimated.

There are essentially three major components of a neural network model: *architecture*, *cost function*, and *search algorithm*. The architecture defines the functional form relating the inputs to the outputs (in terms of network topology, unit connectivity, and activation functions). The search in weight space for a set of weights which minimizes the cost function is the training process. In this paper, we use radial basis function (RBF) networks, and the standard gradient descent is used as a search technique.

The construction of a RBF network involves three layers with entirely different roles. The input layer is made up of source nodes (sensory units) that connect the network to its environment, i.e., low-level feature space. The second layer, the only hidden layer in the network, applies a nonlinear transformation from the input space (low-level feature space) to the hidden space. Generally, the hidden space is of high dimensionality. The hidden layer has RBF neurons, which calculate the hidden layer's net input by combining weighted inputs and biases. The output layer is linear, supplying the block importance given the low-level block representation applied to the input layer. A mathematical justification for the rationale of a nonlinear transformation followed by a linear transformation can be found in [5].

The function learned by RBF networks can be represented by

$$f_i(\mathbf{x}) = \sum_{j=1}^{h} \omega_{ij} G_i(\mathbf{x})$$

where *h* is the number of hidden layer neurons, □*ij* □*R* are the weights. *Gi* is the radial function defined as follows:

$$G_i(\mathbf{x}) = exp(-\frac{\left\|\mathbf{x} - \mathbf{c}_i\right\|^2}{\sigma_i^2})$$

where ***ci*** is the center for *Gi*, and □*i* is the basis function width. The *k*-dimensional mapping can be represented as follows:

$$\mathbf{x} \rightarrow f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_k(\mathbf{x}))$$

where *f* = [*f*1, *f*2, ….., *fk*] is the mapping function.

In summary, the RBF neural network approximates the optimal regression function from feature space to block importance. It is trained off-line with the training samples {**x***i* , **y***i*} (*i* = 1,…, m). For a new block previously unprocessed, its importance can be simply calculated by the regression function *f* given block representation in feature space.

2) Classification by Support Vector Machines

When the labels are discrete numbers, the minimization problem can be regarded as a classification problem. In this section, we describe Support Vector Machines (SVM) which is a pattern classification algorithm developed by V. Vapnik [14]. SVM is based on the idea of *structural risk minimization* rather than *empirical risk minimization*.

We shall consider SVM in the binary classification setting. We assume that we have a data set *ti D* {**x***i* , *yi*} of labeled examples, where *yi* {-1,1}, and we wish to select, among the infinite number of linear classifiers that separate the data, one that minimizes the generalization error, or at least an upper bound on it. V. Vapnik [14] showed that the hyperplane with this property is the one that leaves the maximum margin between the two classes.

Given a new data point **x** to classify, a label is assigned according to its relationship to the decision boundary, and the corresponding decision function is:

$$f(\mathbf{x}) = sign(\sum_{i=1}^{t} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle - b)$$

From this equation it is possible to see that the □*i* associated with the training point **x***i* expresses the strength with which that point is embedded in the final decision function. A remarkable property of this alternative representation is that often only a subset of the points will be associated with non-zero □*i*. These points are called *support vectors* and are the points that lie closest to the separating hyper plane.

The nonlinear support vector machine maps the input variable into a high dimensional (often infinite dimensional) space, and applies the linear support vector machine in the space. Computationally, this can be archived by the application of a (reproducing) kernel.

The corresponding nonlinear decision function is:

$$f(\mathbf{x}) = sign(\sum_{i=1}^{t} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - b)$$

Where *K* is the kernel function. Some typical kernel functions include polynomial kernel, Gaussian RBF kernel, and sigmoid kernel. For multi-class classification problem, one can simply apply one-against-all scheme [6], [7], [12].

We use both the linear SVM and nonlinear SVM with Gaussian RBF kernel to learn the block importance models in our experiments.

VI. EXPERIMENTS

This section provides empirical evidence about the accuracy of the learned block importance models and the factors affecting the learning process.

A. Experiments Setup

The 600 labeled web pages from 405 sites in our user study are used as the dataset in our experiments. Only those blocks, for which at least 3 of the 5 assessors have agreement on their importance, are chosen. As a consequence, totally 4517 blocks are selected from the 4539 labeled blocks.

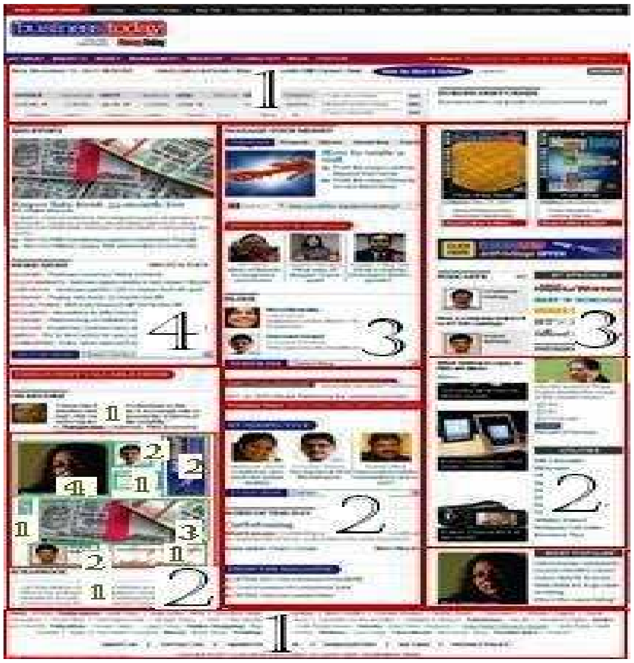

Fig. 3. The tool used in the user study

We randomly split the labeled data into 5 parts and conducted 5- fold cross validation. Classical measures, such as precision, recall, Micro-F1 and Micro-Accuracy (Micro-Acc for short), are borrowed to evaluate the block importance models. For each importance level, precision and recall are reported. And for the overall performance, Micro-F1 and Micro-Acc are provided. In our experiments, Micro-precision, Micro-recall and Micro-F1 are equal since one block can only has one importance value.

In most of our experiments, we divide the importance into 3 levels by combing level 2 and 3. In this section, if not explicated denoted, level 2 refers to the combination of level 2 and 3.

VII. APPLICATIONS OF NESTED BLOCK IMPORTANCE

Block importance plays a significant role in a wide range of web applications. Any application involved web pages will be facilitated by the block importance model, such as information retrieval, web page classification and web adaptation. Why it can benefit such extensive web applications? The essence of its advantages lies in the ability to distinguish the most important content from less important and noisy information. Here we show several promising applications that may take advantage of the block importance model.

Block importance model is motivated by the urge to improve information retrieval performance, thus its direct application lies in this area [1], [10]. Search engine may benefit from block importance in two aspects: one is to improve the overall relevance rank of the returned pages after searching; the other is to locate the important regions carrying more information during searching. For example, in pseudo-relevance feedback technique, the expansion terms will be selected from the region with high importance and block importance could also be combined into term weights.

Another application of block importance is on web page classification [9], [15], [16]. For most of the existing techniques, features used for classification are selected from the whole page. Noisy information in web pages may decrease the accuracy of classification. However, the most useful information and noise could be naturally differentiated by using page segmentation and block importance. In other words, features in important blocks will be chosen or have higher weights than features in unimportant blocks. There also are a few works beginning to explore this interesting topic [5], [15], [16]. Block importance can also be applied to facilitate the web adaptation applications driven by the proliferation of small mobile devices [8]. With the limited display screen sizes of mobile devices, it is a big challenge to provide users the most appealing information. Block importance could be used to effectively decide which parts of the pages should be first displayed at the screen and hence satisfy users' information needs to the largest degree. There are many other applications may take advantages of the block importance model. We just name a few here. When web pages are segmented and importance is automatically assigned to the blocks, we have a powerful tool to enhance traditional techniques and create new techniques.

VIII. CONCLUSION

The explosive growth of information on the Web makes it critical to develop techniques to distinguish importance information from unimportant one. Similar to methods of identifying authoritative web pages on the Web, we introduce a way to identify important portions within web pages. We view this as a learning problem and aim to find functions to describe the correlations between web page blocks and importance values. The VIPS algorithm is used to partition a web page into multiple semantic blocks and features are extracted from each block. Then learning algorithms, such as SVM and neural network, are applied to train block importance models based on the features. In our experiments, the best model can achieve Micro-F1 79% and Micro-Accuracy 85.9% on block importance assignment, which is quite close to a person's performance. Although spatial features have major effects on block importance, better performance can be achieved by integrating content features. For different kinds of spatial features, the window spatial features proved to be the most effective one. Our work showed that, just like our user study demonstrated, people do have consistent opinions about the importance of blocks in web pages and accurate models can be built to deduce the importance values automatically.

REFERENCES

[1] Bar-Yossef, Z. and Rajagopalan, S., *Template Detection via Data Mining and its Applications,* In the proceedings of Eleventh World Wide Web conference (WWW 2002), May 2002.

[2] Brin, S. and Page L., *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, In the Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, 1998.

[3] Cai, D., Yu, S., Wen, J.-R. and Ma, W.-Y., *VIPS: a visionbased page segmentation algorithm*, Microsoft Technical Report, MSR-TR-2003-79, 2003.

[4] Chen, J., Zhou, B., Shi, J., Zhang, H.-J. and Qiu, F.,*Function-Based Object Model Towards Website Adaptation*, In the pro ceedings of the Tenth World Wide Web conference (WWW10), Budapest, Hungary, May 2001.

[5] Cover, T. M. *Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition*, IEEE Transactions on Electronic Computers, vol. EC-14, pp. 326-334.

[6] Dietterich, T. G. and Bakiri, G., *Solving multiclass learning problem via error correcting output codes,* Journal of Artificial Intelligence Research, 2:263-286, 1995.

[7] Dietterich, T.G. and Bakiri, G., *Error-correcting output codes: a general method for improving multiclass inductive learning programs,* In the proceedings of AAAI-91, pages 572-577. AAAI press / MIT press, 1991.

[8] Gupta, S., Kaiser, G., Neistadt, D. and Grimm, P., *DOMbasedContent Extraction of HTML Documents,* In the proceedings of the Twelfth World Wide Web conference (WWW 2003), Budapest, Hungary, May 2003.

[9] Kovacevic, M., Diligenti, M., Gori, M. and Milutinovic, V., *Recognition of Common Areas in a Web Page Using VisualInformation: a possible application in a page classification,*in the proceedings of 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan, December,
2002.

[10] Lin, S.-H. and Ho, J.-M., *Discovering Informative ContentBlocks from Web Documents,* In the proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD'02), 2002.

[11] Liu, H., Xie, X., Ma, W.-Y. and Zhang, H.-J., *Automatic Browsing of Large Pictures on Mobile Devices,* in the proceedings of 11th ACM International Conference on Multimedia, Berkeley, CA, USA, Nov. 2003

[12] Mayoraz, E. and Alpaydin, E., *Support vector machines for multiclass classification,* in the proceedings of the international workshop on artificial intelligence neural networks, 1999.

[13] Schwartz, M., and Task Force on Bias-Free Language. Guidelines for Bias-Free Writing. Indiana University Press, Bloomington, IN, 1995

[14] V. Vapnik. *Principles of risk minimization for learning theory.* In D. S. Lippman, J. E. Moody, and D. S. Touretzky, editors, Advances in Neural Information Processing Systems 3, pages 831-838. Morgan Kaufmann, 1992.

[15] Yi, L. and Liu, B., *Web Page Cleaning for Web Mining through Feature Weighting,* in the proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August, 2003.

[16] Yi, L. and Liu, B., *Eliminating Noisy Information in WebPages for Data Mining,* In the proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2003), Washington, DC, USA, August, 2003.

[17] Yu, S., Cai, D., Wen, J.-R. and Ma, W.-Y., *Improving Pseudo-Relevance Feedback in Web Information retrieval Using Web Page Segmentation,* In the proceedings of Twelfth World Wide Web conference (WWW 2003), Budapest, Hungary, May 2003.

U. Vinod Kumar, Working as Assistant Professor in the Department of CSE, Regency Institute of Technology, Yanam, Currently he is working in data mining.

Shaik Yacoob, Working as Assistant Professor in the Department of CSE, Regency Institute of Technology, Yanam, Currently he is working in Software engineering..

Sudam Sekhar Panda, Working as Sr. Assistant Professor in the Department of CSE, Regency Institute of Technology, Yanam, Currently he is working in image processing.

Ch. Rajaramesh, Working as Associate Professor in the Department of CSE, Regency Institute of Technology, Yanam, Currently he is working in data mining.