Learning Block Importance Models for Web Pages

Ruihua Song, Haifeng Liu[1*], Ji-Rong Wen, Wei-Ying Ma
Microsoft Research Asia, 49 Zhichun Road, Beijing, 100080, P.R. China
{i-rsong, jrwen, wyma}@microsoft.com
[1]Department of Computer Science, University of Toronto, Toronto, ON, Canada
[1]hfliu@cs.toronto.edu

ABSTRACT

Previous work shows that a web page can be partitioned into multiple segments or blocks, and often the importance of those blocks in a page is not equivalent. Also, it has been proven that differentiating noisy or unimportant blocks from pages can facilitate web mining, search and accessibility. However, no uniform approach and model has been presented to measure the importance of different segments 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 blocks in a web page. We define the block importance estimation as a learning problem. First, we use a vision-based page segmentation algorithm to partition a web page into semantic blocks with a hierarchical structure. Then spatial features (such as position and size) and content features (such as the number of images and links) are extracted to construct a feature vector for each block. Based on these features, learning algorithms are used to train a model to assign importance to different segments in the web page. In our experiments, the best model can achieve the performance with Micro-F1 79% and Micro-Accuracy 85.9%, which is quite close to a person's view.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *Data Mining*; I.7.m [**Document and Text Processing**]: Miscellaneous; H.5.1 [**Information Systems Applications**]: Information Interfaces and Presentation; H.4.3 [**Information Systems Applications**]: Communications Applications – *Information Browsers*;

General Terms

Algorithms, Human Factors.

Keywords

Block importance model, page segmentation, web mining, classification.

1. INTRODUCTION

The Web provides people a convenient media to disseminate information. With the rapid increase of information on the Web, an effective method for users to discern the useful information from the non-useful information is urgently required. 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 PageRank [2]. However, up to date, there is no effective technique for the latter aspect. Most techniques consider the whole web page as an atomic unit and treat different portions in a web page equally.

Obviously, the information in a web page is not equally important. For example, consider the web page in Figure 1, the headline in a news web site is much more attractive to users than the navigation bar. Moreover, 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 values for different segments in the web page.

To distinguish different information in a web page, we first need to segment a web page into a set of blocks. There are several kinds of methods for web page segmentation. The most popular ones are DOM-based segmentation [4], location-based segmentation [9] and Vision-based Page Segmentation (VIPS) [17][3]. These methods are distinguished from one another 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 block importance model in this paper to assign importance values to different blocks in a page. First, the Vision-based Page Segmentation (VIPS) algorithm is used to partition a web page into blocks according to the content coherence by analyzing the visual layout of the page. Then, the block features (including spatial features and content features) are extracted to represent the blocks. Finally, based on these features, we use Support Vector Machines (SVM) and neural network methods to learn general block importance models.

The main contributions of our work are:

1. A comprehensive user study is conducted to validate that people do have consistent opinions on the importance of different regions in web pages.
2. A block importance model is proposed to automatically assign importance weights to different regions in a web page. This model takes into account spatial features and content features.

WWW 2004, May 17-22, 2004, New York, NY USA.
ACM 1-58113-844-X/04/0005.

[*] The work presented in this paper was carried out at Microsoft Research Asia.

3. Two methods, based on neural network and SVM for importance assignment, are proposed.

4. Many promising applications of the block importance model are discussed.

The rest of the paper is organized as follows. In Section 2, related work is described. In Section 3, we introduce the page segmentation methods, especially the VIPS method. Section 4 is the user study we conducted to validate that people do have consistent opinions about the importance of different regions in web pages. In Section 5, we introduce the details of the block importance model, including the features and learning methods we use. Experimental evaluation is presented in Section 6 to assess the performance of our model. Finally, we discuss applications of our work in Section 7 and draw a conclusion.

Figure 1. A sample web page containing multiple segments with different importance

2. RELATED WORK

Related work 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 approaches 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 experiments 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, InfoDiscover, 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 web site like www.msn.com which is full of links, the rule would remove almost all useful contents. Besides 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 approach is close to ours. Since it is rule-based, it cannot deal with dozens of features with complicated correlations.

3. 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 a web page. 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 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. VIPS makes full use of page layout features such as font, color and size. It first extracts all the suitable nodes from the HTML DOM tree, and then finds the separators between these nodes. Here, separators denote the horizontal or vertical lines in a web page that visually do not cross any node. Based on these separators, the semantic tree of the web page is constructed. A value called degree of coherence (DOC) is assigned for each node to indicate how coherent it is. Consequently, VIPS can efficiently keep related content together while separating semantically different blocks from each other.

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, and all leaf nodes consist of a flat segmentation of a web page. The granularity of segmentation in VIPS is controlled by a predefined degree of coherence (PDOC), which plays a role as a threshold of the most appropriate granularity for different applications. The segmentation only stops when the DOCs of all blocks are no smaller than the PDOCs. Figure 2 shows the result of using VIPS to segment a sample CNN web page. For details of the VIPS algorithm, please refer to [3].

Figure 2. VIPS segmentation of a sample web page

4. A USER STUDY OF BLOCK IMPORTANCE

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 concept. It means the concentration of mental energy 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 difficult to find a general model describing such subjective importance definitions.

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 not the user's views.

We conduct a user study to validate that such an objective importance model does exist. The tool used in the study is illustrated in Figure 3. First, a web page is segmented into a hierarchical block structure 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 leave blocks form a partition of the page.

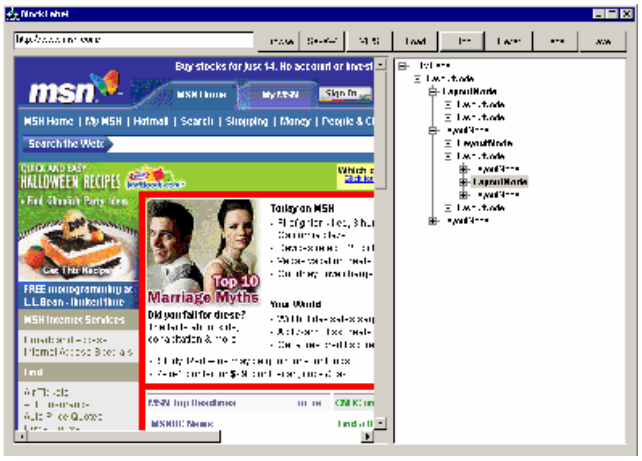

Figure 3. The block importance labeling tool for conducting the user study

We downloaded 600 web pages from 405 sites in 3 categories in yahoo: news, science and shopping. Each category includes 200 pages. We treat the homepage and inner pages of a website as different pages, thus the impact of websites is ignored here. However, we have checked the downloaded web pages to try to collect pages with diverse layouts and contents. After page segmentation, we obtained a total of 4539 blocks.

We then asked 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.

The labeling process is independent among assessors. No one could see the labeling results of others. All of the assessors are graduated students of computer science and they have good knowledge of both English and Chinese.

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% blocks (see Table 1).

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

Levels	3/5 agreement	4/5 agreement	5/5 agreement
1,2,3,4	0.929	0.535	0.237
1,(2,3),4	0.995	0.733	0.417
(1,2,3),4	1	0.932	0.828

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 difficult to discern for the assessors while level 4 can be most clearly identified. Accordingly, when levels 1, 2, 3 are merged, the assessors reached very high agreement (Table 3). Moreover, when (1, 4) and (2, 3) are merged to 2 levels, the consistency is also quite good. The reasons may lie in that most important blocks and most unimportant blocks can be more easily distinguished from those that lie in between, and levels 2 and 3 are the most blurry zones to be distinguished. So, in practice, we combine levels 2 and 3.

The user study clearly demonstrated 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.

Table 2: Agreement on all kinds of 3-level importance

Levels	3/5 agreement	4/5 agreement	5/5 agreement
(1,2),3,4	0.965	0.76	0.562
1,(2,3),4	0.995	0.733	0.417
1,2,(3,4)	0.963	0.614	0.318
(1,3),2,4	0.965	0.553	0.244
1,3,(2,4)	0.965	0.555	0.248
(1,4),2,3	0.934	0.539	0.24

5. BLOCK IMPORTANCE MODEL

Web page designers tend to organize their content in a reasonable way: giving prominence 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:

$$< block\ features > \rightarrow block\ importance$$

Table 3: Agreement on all kinds of 2-level importance

Levels	3/5 agreement	4/5 agreement	5/5 agreement
(123),4	1	0.932	0.828
1,(2,3,4)	1	0.808	0.568
(1,3,4),2	1	0.637	0.332
(1,2,4),3	1	0.786	0.582
(1,2),(3,4)	1	0.736	0.42
(1,4),(2,3)	1	0.838	0.644
(1,3),(2,4)	1	0.573	0.255

5.1 Block Features

Let us take a look at the web page in Figure 1 again and see what features are used to differentiate the important parts from unimportant parts. Typically, web designers would put the most important information in the center and put the navigation bar on the header or the left side and the copyright on the footer (the information in the solid circles is more important than those in the dashed circle in Figure 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 Figure 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 the model.

5.1.1 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:

{BlockCenterX, BlockCenterY, BlockRectWidth, BlockRectHeight}

BlockCenterX and *BlockCenterY* are the coordinates of the center point of the block and *BlockRectWidth,BlockRectHeight* 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 them into *relative spatial features*, as given below:

{BlockCenterX/PageWidth, BlockCenterY/PageHeight, 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 Figure 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 Figure 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 modify the relative spatial features into *window spatial features*. Instead of using the height of the whole page for normalization, we use a fixed-height window instead.

$$BlockRectHeight = BlockRectHeight / WindowHeight;$$

Also, feature BlockCenterY is modified as:

$$BlockCenterY = \begin{cases} BlockCenterY / (2 * HeaderHeight); \\ \qquad if \quad BlockCenterY < HeaderHeight \\ 0.5; \qquad if \quad HeaderHeight < BlockCenter \\ \qquad Y < PageHeight - FooterHeight \\ 1 - (PageHeight - BlockCenterY) / (2 * FooterHeight); \\ \qquad otherwise \end{cases}$$

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

5.1.2 Content features

The following 9 features are used to represent the content of a block:

{ImgNum, ImgSize, LinkNum, LinkTextLength, InnerTextLength, InteractionNum, InteractionSize, FormNum, FormSize}

ImgNum and *ImgSize* are the number and size of images contained in the block. *LinkNum* and *LinkTextLength* are the number of hyperlinks and anchor text length of the block. *InnerTextLength* is the length of text between the start and end tags of HTML objects. *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 related to the importance. For example, an advertisement may contain only images but no 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 *LinkNum* of a block is normalized by the link number of the whole page.

5.2 Learning 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 approaches addressing the problem of block function identification. In [4], an automatic rule-based approach is presented to detect the functional property and category of objects. 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 approach, that is, learning from examples. Specially, some blocks are pre-labeled by several people and thus each labeled block can be represented as ($\mathbf{x}$, y) where $\mathbf{x}$ 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, the problem becomes to find a function *f* such that

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

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 treating it as a regression problem. Another is the SVM learning method when viewing it as a classification problem.

5.2.1 Regression by Neural Network

When the labels are continuous real numbers, 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 the 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, $\varpi_{ij} \in R$ are the weights. G_i is the radial function defined as follows:

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

where $\boldsymbol{c}_i$ is the center for G_i, and σ_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 $f = [f_1, f_2, \ldots, f_k]$ 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 $\{\mathbf{x}_i, \mathbf{y}_i\}$ (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.

5.2.2 *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 [13]. 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 $D = \{\mathbf{x}_i, y_i\}_{i=1}^t$ of labeled examples, where $y_i \in \{-1,1\}$, 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 [13] showed that the hyperplane with this property is the one that leaves the maximum margin between the two classes. Given a new data point $\mathbf{x}$ 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 α_i associated with the training point $\mathbf{x}_i$ 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 α_i. These points are called *support vectors* and are the points that lie closest to the separating hyperplane.

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.

Based on the learned block importance model, we implemented a browser to display web pages with importance labeling. Figure 4 shows an example. When a page is loaded into the browser, the page is segmented first by the VIPS algorithm and then an importance value is calculated for each block based on the block importance model.

Figure 4. Web page browser with automatic block importance labeling. Blocks with importance level 4, level 2 and level 1 are framed with colors red, green and blue, respectively. Note that we superimpose the level number on the page to make the result easy to see.

6. EXPERIMENTS

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

6.1 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 agreed on their importance values, are chosen. Consequently, a total of 4517 blocks are selected from the 4539 labeled blocks.

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) [14], are used 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 have 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 explicitly stated, level 2 refers to the combination of level 2 and 3.

6.2 Comparison of Learning Methods

Three learning methods, linear SVM, nonlinear SVM with RBF kernel and a RBF network are used to learn the models. The best performance obtained by these methods are reported and compared in Table 4. SVM with RBF kernel achieved the best performance with Micro-F1 79% and Micro-Acc 85.9%. The linear SVM performed worse than both SVM with RBF kernel and RBF network. The results indicate that a nonlinear combination of the features is better than a linear combination.

Table 4: Comparison of learning methods

Methods	Level 1	Level 2	Level 4	Micro-F1	Micro-Acc
SVM (RBF)	0.763 (P)	0.796 (P)	0.839 (P)	0.790	0.859
	0.776 (R)	0.804 (R)	0.770 (R)		
SVM (linear)	0.664 (P)	0.719 (P)	0.865 (P)	0.716	0.811
	0.656 (R)	0.762 (R)	0.693 (R)		
RBF network	0.716 (P)	0.772 (P)	0.819 (P)	0.757	0.838
	0.766 (R)	0.762 (R)	0.714 (R)		

Table 5: Comparison between 4-level and 3-level block importance models (SVM with RBF kernel)

	Level 1	Level 2	Level 3	Level 4	Micro-F1	Micro-Acc
4-level	0.708 (P)	0.643 (P)	0.567(P)	0.826 (P)	0.685	0.843
	0.782 (R)	0.658 (R)	0.372(R)	0.822 (R)		
3-level	0.763 (P)		0.796 (P)	0.839 (P)	0.790	0.859
	0.776 (R)		0.804 (R)	0.770 (R)		

6.3 3-level Importance vs. 4-level Importance

As mentioned in the user study section, when combing level 2 and 3, more consistent labeling results of block importance could be obtained. Based on the 3-level importance labeling and 4-level importance labeling, we train two block importance models, respectively, by using the SVM with RBF kernel method. Table 5 shows the performance of the two models. For the 4-level importance model, it is clear to see that the precision and recall of level 2 and 3 are not good. By combining them, the precision and recall at level 2 in the 3-level model are increased significantly. As a consequence, the Micro-F1 and Micro-Acc of 3-level model is better than 4-level model.

6.4 Spatial Features vs. All Features

To measure the impacts of spatial features and content features respectively, we also build a model which only uses spatial features to represent blocks. We also use SVM with RBF kernel to train the model. Table 6 compares the performance of the model with the one using all features. It is not surprising to see that the model with only spatial features can achieve good performance. When content features are added, there is a significant increase in performance. It proves that content features do provide complementary information to spatial features to measure block importance.

Table 6: Comparison between 4-level and 3-level block importance models (SVM with RBF kernel)

	Level 1	Level 2	Level 4	Micro-F1	Micro-Acc
Spatial	0.714 (P)	0.754 (P)	0.805 (P)	0.748	0.832
	0.684 (R)	0.769 (R)	0.841 (R)		
All	0.763 (P)	0.796 (P)	0.839 (P)	0.790	0.859
	0.776 (R)	0.804 (R)	0.770 (R)		

6.5 Comparison of Different Spatial Features

Window spatial features are used in the above experiments. Here we compare all of the three kinds of spatial features: absolute, relative and window. Three block importance models are trained based on the three kinds of spatial features, respectively. The result comparison is listed in Table 7. The performance of the model using absolute spatial features is much worse that the other two. And the performance of the model using window spatial features is slightly higher than the one using relative spatial features. This performance improvement mainly comes from long pages. Since short pages are dominant in our labeled data, the overall performance improvement is not very big.

Table 7: Comparison of three kinds of spatial features

Features	Level 1	Level 2	Level 4	Micro-F1	Micro-Acc
Absolute	0.814 (P)	0.534 (P)	1 (P)	0.543	0.695
	0.060 (R)	0.990 (R)	0.018 (R)		
Relative	0.727 (P)	0.791 (P)	0.894 (P)	0.777	0.854
	0.768 (R)	0.788 (R)	0.760 (R)		
Window	0.763 (P)	0.796 (P)	0.839 (P)	0.790	0.859
	0.776 (R)	0.804 (R)	0.770 (R)		

6.6 Block Importance Model vs. Human Assessors

Finally, we compare the performance of our learned model with those human assessors. Since we apply a voting mechanism to determine the labeled importance of blocks, even the assessors may not achieve a 100% Micro-F1. We calculate the labeling

performance for the 5 assessors and compare their performance with our model. The results show that the performance of our model is quite close to that of a human (Table 8).

Table 8: Block Importance Model vs. Human Assessors

	Level 1	Level 2	Level 3	Micro-F1	Micro-Acc
Assessor 1	0.817 (P)	0.871 (P)	0.934 (P)	0.858	0.906
	0.856 (R)	0.857 (R)	0.871 (R)		
Assessor 2	0.756 (P)	0.815 (P)	0.816 (P)	0.792	0.861
	0.834 (R)	0.782 (R)	0.715 (R)		
Assessor 3	0.864 (P)	0.838 (P)	0.852 (P)	0.849	0.899
	0.815 (R)	0.881 (R)	0.809 (R)		
Assessor 4	0.904 (P)	0.797 (P)	0.827 (P)	0.830	0.887
	0.684 (R)	0.908 (R)	0.912 (R)		
Assessor 5	0.849 (P)	0.895 (P)	0.938 (P)	0.882	0.921
	0.924 (R)	0.882 (R)	0.762 (R)		
Average	0.838 (P)	0.843 (P)	0.873 (P)	0.842	0.895
	0.823 (R)	0.862 (R)	0.814(R)		
Our model	0.763 (P)	0.796 (P)	0.839 (P)	0.790	0.859
	0.776 (R)	0.804 (R)	0.770 (R)		

7. APPLICATIONS OF BLOCK IMPORTANCE

Block importance can play a significant role in a wide range of web applications. Any application involving web page analysis, such as information retrieval, web page classification and web adaptation, could benefit from the block importance model. The essence of this model's advantages lies in its ability to distinguish the most important content from less important and noisy information. Here we show a few promising applications that may take advantage of our block importance model.

The study of block importance model is mainly motivated by the urge to improve Web information retrieval performance, thus its direct application lies in the area [1][10]. Web information retrieval may benefit from block importance in three aspects. The first one is to improve the relevance rank of the returned web pages. For example, words in important blocks could be weighted higher than those in less important blocks and noisy contents in pages could be filtered out in advance. Another one is to improve link analysis algorithms, such as PageRank [2]. In traditional methods, links with mixed topics in a page are treated as a whole and weighted equally, while recommendation relationships that these links imply are not consistent or equal. With block importance, links can be differentiated naturally and assigned different weights so that page importance could be spread more precisely. Finally, block importance could be leveraged to improve the presentation of search results. For example, sentences in important blocks could be chosen to produce better snippets of web pages.

Another application of block importance is for 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 have been a few approaches beginning to explore this topic [5][15][16].

Block importance can also be applied to facilitate 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 on the screen and hence satisfy users' information needs to the largest possible degree.

There are many other applications that may take advantage 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.

8. CONCLUSION

The explosive growth of information on the Web makes it critical to develop techniques to distinguish important 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. Among 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 a web page and effective models can be built to deduce the importance values automatically.

9. ACKNOWLEDGEMENTS

We are grateful to Professor H-Arno Jacobsen at University of Toronto for his valuable comments and revision on our paper.

10. REFERENCES

[1] Bar-Yossef, Z. and Rajagopalan, S., *Template Detection via Data Mining and its Applications,* in the proceedings of 11th 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 vision-based 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 proceedings of the 10th 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, i*n the proceedings of AAAI-91, pages 572-577. AAAI press / MIT press, 1991.

[8] Gupta, S., Kaiser, G., Neistadt, D. and Grimm, P., *DOM-based Content Extraction of HTML Documents,* in the proceedings of the 12th 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 Visual Information: 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 Content Blocks 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] 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

[14] Yang, Y., *An Evaluation of Statistical Approaches to Text Categorization,* Information Retrieval, Vol. 1, Number 1-2, pp.69-90, 1999

[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 Web Pages 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.