Improving Pseudo-Relevance Feedback in Web Information Retrieval Using Web Page Segmentation

Shipeng Yu[1*] Deng Cai[2*] Ji-Rong Wen[*] and Wei-Ying Ma[*]

[*]Microsoft Research Asia
Beijing, China
{jrwen, wyma}@microsoft.com

[1]Peking University
Beijing, China
ysp@is.pku.edu.cn

[2]Tsinghua University
Beijing, China
caideng00@mails.tsinghua.edu.cn

ABSTRACT

In contrast to traditional document retrieval, a web page as a whole is not a good information unit to search because it often contains multiple topics and a lot of irrelevant information from navigation, decoration, and interaction part of the page. In this paper, we propose a **VI**sion-based **P**age **S**egmentation (VIPS) algorithm to detect the semantic content structure in a web page. Compared with simple DOM based segmentation method, our page segmentation scheme utilizes useful visual cues to obtain a better partition of a page at the semantic level. By using our VIPS algorithm to assist the selection of query expansion terms in pseudo-relevance feedback in web information retrieval, we achieve 27% performance improvement on Web Track dataset.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Relevance feedback*; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia

General Terms

Algorithms, Performance, Human Factors

Keywords

Web information retrieval, page segmentation, query expansion, relevance feedback

1. INTRODUCTION

To cope with the information explosion of the Web, effective and efficient information retrieval has been the most challenging problem for a search engine. Pseudo-relevance feedback, also known as local feedback or blind feedback, is a technique commonly used to improve retrieval performance [3, 10]. Its basic idea is to extract expansion terms from the top-ranked documents to formulate a new query for a second round retrieval. Through a query expansion, some relevant documents missed in the initial round can then be retrieved to improve the overall performance. Clearly, the effect of this method strongly relies on the quality of selected expansion terms. If the words added to the original query are unrelated to the topic, the quality of the retrieval is likely to be degraded. Since the Web is a highly volatile and heterogeneous information source and contains a lot of low-quality documents [1], such a naive pseudo-relevance feedback is not capable of producing a satisfactory result.

There are many reasons why simple pseudo-relevance feedback does not work. First, web pages usually do not contain pure content. A web page typically contains various types of materials that are not related to the topic of the web-page, such as:

- ✧ Navigation – intra and inter hyperlinks to guide the user to different part of a web page.
- ✧ Decoration – pictures, animations, logos and so forth for attraction or advertisement purposes.
- ✧ Interaction – forms to collect user information or provide searching services.
- ✧ Other special words or paragraphs such as copyrights and contact information.

In terms of query matching and term weighting, all of the above stuffs are considered noises and harmful to retrieval performance. Secondly, a web page usually contains multiple topics, for example, a news page containing many different comments on a particular event or politician and a conference web page containing sponsors from different companies and organizations. Although traditional documents also often have multiple topics, they are less diverse so that the impact on retrieval performance is smaller.

For pseudo-relevance feedback, the quality of expansion terms is heavily affected by the top-ranked documents. Noise and multi-topics are two major negative factors for expansion term selection. For instance, if an advertisement is embedded in a top-ranked web page at the first round, then some terms from the advertisement may be selected as expansion terms. Once these terms are used to expand the query for the second round retrieval, irrelevant web pages containing these advertisement terms could be ranked highly. Similarly, for a web page containing multiple topics, the selected terms are also subject to this uncertainty which may decrease the retrieval performance. Therefore, it is necessary to segment a web page into semantically related units so that noisy information can be filtered out and multiple topics can be distinguished.

The structure of a HTML page can be represented as a tag tree by DOM (Document Object Model, see http://www.w3.org/DOM/). DOM-based segmentation approaches are widely used these years for record boundary discovery [4, 11], topic distillation [7] and adaptive content delivery [5, 8, 13]. 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. To the best of our knowledge, few works are done on applying DOM-based page segmentation methods to improve web information retrieval. Some preliminary studies are performed in [9, 14] but the results are not encouraging. 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 [12], and is often not accurate enough to discriminate different semantic blocks in a web page.

To segment a web page into semantically related units, the visual presentation of the page contains a lot of useful cues. Generally, a web page designer would organize the content of a web page to make it easy for reading. Thus, semantically related content is usually grouped together and the entire page is divided into regions for different contents using explicit or implicit visual separators such as lines, blank areas, images, font sizes, colors, etc [18]. Our goal is to derive this content structure from the visual presentation of a web page. In this paper, we propose VIPS (**VI**sion-based **P**age **S**egmentation) algorithm to segment a web page. The algorithm makes full use of the layout features of the page and tries to partition the page at the semantic level. Each node in the extracted content structure will correspond to a block of coherent content in the original page.

Compared with DOM based methods, the segments obtained by VIPS are much more semantically aggregated. Noisy information, such as navigation, advertisement, and decoration can be easily removed because they are often the blocks in some fixed region of a page. Content with different topics is distinguished as separate blocks. With the assumption that terms in different segments are not correlated to a common topic, the expansion term selection is constrained within a few segments instead of the whole web page. By combining VIPS algorithm with the pseudo-relevance feedback method, we propose a novel segment-based pseudo-relevance feedback method for web information retrieval. The experimental results in Section 4 prove that our proposed method can significantly improve the retrieval performance, both in terms of precision and recall.

The rest of the paper is organized as the following. The details of the VIPS algorithm are introduced in Section 2. In Section 3, we illustrate how to utilize web page segmentation to improve pseudo-relevance feedback. The experimental results are reported in Section 4. Section 5 summarizes our contributions and concludes the paper.

2. THE VIPS ALGORITHM

People view a web page through a web browser and get a 2-D presentation which provides many visual cues to help distinguish different parts of the page. Examples of these cues include lines, blanks, images, different font sizes, and different colors, etc. For the purpose of easy browsing and understanding, a closely packed region within a web page is usually about a single topic. This observation motivates us to segment a web page from its visual presentation.

In [18] and [8], some visual cues are used in DOM analysis. They try to identify the logical relationships within a web page based on visual layout information, but these approaches rely too much on the DOM structure. Gu [12] tries to construct a web content structure by breaking out the DOM tree and comparing similarity among all the basic DOM nodes. Since a normal web page may have hundreds of basic elements, the algorithm is time-consuming and inflexible, not suitable to deal with a large amount of web pages.

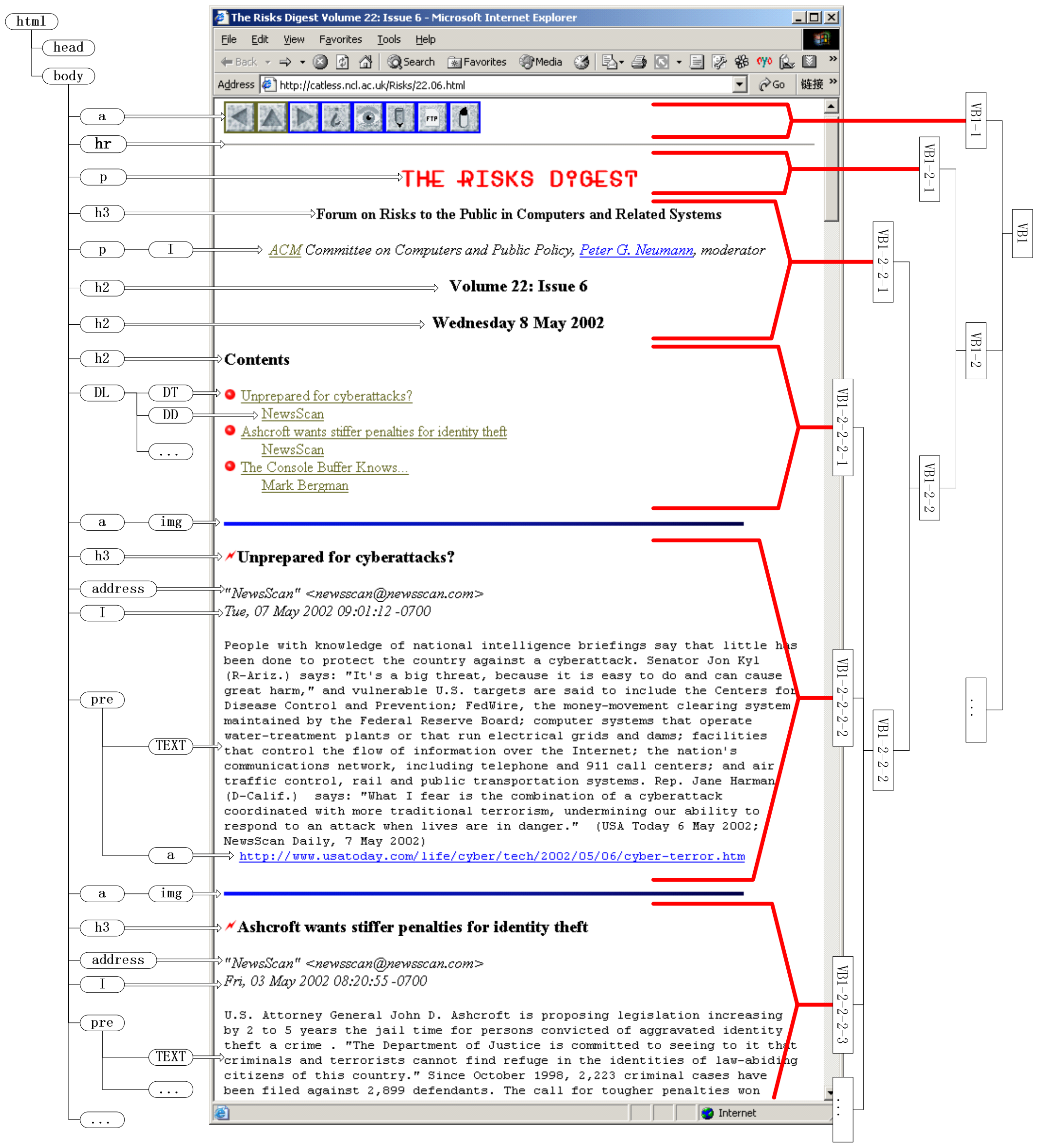
We propose a vision-based web page segmentation algorithm by combining the DOM structure and visual cues. After the segmentation process, a tree-like content structure of the page based on its visual presentation is constructed. Take the web page in Figure 1 as an example (this web page can be accessed at http://catless.ncl.ac.uk/Risks/22.06.html with a little modification for simplicity). The tree on the left side is the DOM structure of the page. Since tags are distributed within the *BODY* tag without any hierarchies, it is hard to directly extract segments from the DOM structure that correspond to a visual block and contain a unit of information in the page. However from the presentation style, we can easily distinguish each part from the others and recognize the different topic in the page, as illustrated on the right side of the page.

2.1 Vision-Based Content Structure for Web Pages

The *vision-based content structure* of a web page is a tree-like structure. Each node of the tree represents a region in the web page, which we call a *visual block* or *vb* for abbreviation. If a node has children in the structure, visual blocks of the children are *contained* within that of the node and form a *partition* of it. As shown in right side of Figure 1, the visual blocks *VB1-2-1* and *VB1-2-2* are children of *VB1-2* and act as a partition in the page.

For each node, the *Degree of Coherence* (*DoC)* is defined to show how coherent it is. The value of *DoC* usually ranges from 0 to 1. We can pre-define the *Permitted Degree of Coherence* (*PDoC*） to achieve different granularities of page segmentation for different applications. The less the *PDoC* is, the coarser the content structure would be. For example, in Figure 1, the visual block *VB1-2-2-2-2* may not be further partitioned with an appropriate *PDoC.*

The vision-based content structure is more likely to provide a semantic partitioning of the page. Every node of the structure, especially the leaf node, is more likely to convey a certain semantic meaning and help to build a higher semantic via the hierarchy. For instance, in Figure 1 we can say that *VB1-2-1* denotes the title of the page, *VB1-2-2-2-1* shows table of contents and *VB1-2-2-2-2* tells a whole story.



THE RISKS DIGEST

Forum on Risks to the Public in Computers and Related Systems

ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator

Volume 22: Issue 6

Wednesday 8 May 2002

Contents

- Unprepared for cyberattacks?
  NewsScan
- Ashcroft wants stiffer penalties for identity theft
  NewsScan
- The Console Buffer Knows...
  Mark Bergman

Unprepared for cyberattacks?

"NewsScan" <newsscan@newsscan.com>
Tue, 07 May 2002 09:01:12 -0700

```
People with knowledge of national intelligence briefings say that little has
been done to protect the country against a cyberattack. Senator Jon Kyl
(R-Ariz.) says: "It's a big threat, because it is easy to do and can cause
great harm," and vulnerable U.S. targets are said to include the Centers for
Disease Control and Prevention; FedWire, the money-movement clearing system
maintained by the Federal Reserve Board; computer systems that operate
water-treatment plants or that run electrical grids and dams; facilities
that control the flow of information over the Internet; the nation's
communications network, including telephone and 911 call centers; and air
traffic control, rail and public transportation systems. Rep. Jane Harman
(D-Calif.)  says: "What I fear is the combination of a cyberattack
coordinated with more traditional terrorism, undermining our ability to
respond to an attack when lives are in danger."  (USA Today 6 May 2002;
NewsScan Daily, 7 May 2002)
  http://www.usatoday.com/life/cyber/tech/2002/05/06/cyber-terror.htm
```

Ashcroft wants stiffer penalties for identity theft

"NewsScan" <newsscan@newsscan.com>
Fri, 03 May 2002 08:20:55 -0700

```
U.S. Attorney General John D. Ashcroft is proposing legislation increasing
by 2 to 5 years the jail time for persons convicted of aggravated identity
theft a crime . "The Department of Justice is committed to seeing to it that
criminals and terrorists cannot find refuge in the identities of law-abiding
citizens of this country." Since October 1998, 2,223 criminal cases have
been filed against 2,899 defendants. The call for tougher penalties won
```

Figure 1. A sample web page with its DOM structure on the left and our extracted vision-based content structure on the right

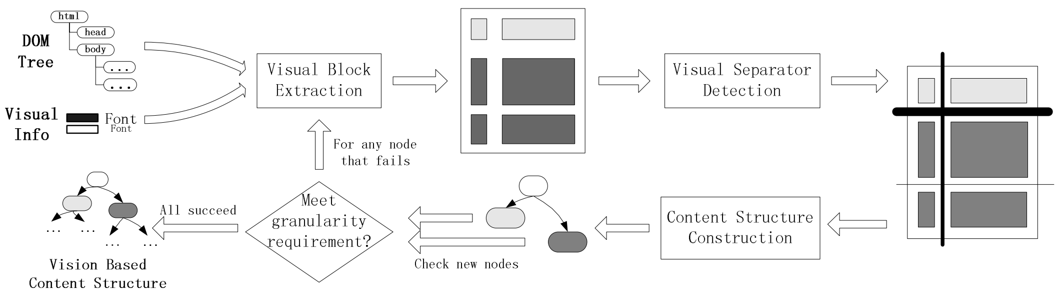


Figure 2. Flowchart of the VIPS algorithm

2.2 Vision-based Page Segmentation Algorithm

To obtain the vision-based content structure for a web page, we introduce the VIPS (Vision-based Page Segmentation) algorithm which makes use of DOM structure as well as visual cues. The flowchart of the segmentation process is illustrated in Figure 2.

First, by calling the analyzer embedded in the web browser, we obtain the DOM structure and visual information such as position, color, font size, font weight, etc. All the visual information comes from HTML elements and attributes. From the subtree within *BODY* in the DOM structure, we start the following iterative process to build the content structure.

Step 1: Visual Block Extraction

In this phase, we aim at finding all appropriate visual blocks contained in the current subtree. Normally, every node inside a current node can represent a visual block, like all the children of *BODY* in Figure 1. However, some "huge" nodes such as *TABLE* and *P* may act only for organization purpose and are not appropriate to represent a single visual block. Therefore, in these cases we should further divide the current node and replace it by its children. This process is iterated until all appropriate nodes are found to represent the visual blocks in the web page.

At the end of this step, for each node that represents a visual block, its *DoC* value is set according to its intra visual difference. Some heuristic rules are used to determine whether a DOM node should be replaced as the following:

- ✧ Tag cue: Tags such as *HR* are often used to separate different topics from visual perspective. Therefore we prefer to divide a DOM node if it contains these tags.
- ✧ Color cue: We divide a DOM node if its background color is different from one of its children's.
- ✧ Text cue: If most of the children of a DOM node are Text nodes (i.e., no tags surround them), we do not divide it.
- ✧ Size cue: We prefer to divide a DOM node if the standard deviation of size of its children is larger than a threshold.

In Figure 1, the *BODY* tag will be divided according to Tag cue and Size cue. Among all of its children, node *PRE* will not be divided according to Text cue.

Step 2: Visual Separator Detection

When all blocks are extracted, they are put into a pool for separator detection. We define *visual separators* as horizontal or vertical lines in a web page that visually cross with no blocks in the pool. We set appropriate weight to each separator according to the following patterns and select those with highest weight as the actual separators.

- ✧ Distance pattern: The more the distance between blocks on different side of the separator, the higher the weight.
- ✧ Tag pattern: If a visual separator is at the same position as some tags such as *HR*, its weight is made higher.
- ✧ Font pattern: If differences of font properties such as font size and font weight are more clearly on two sides of the separator, the weight is made higher.
- ✧ Color pattern: If background colors are different on two sides of the separator, the weight is made higher.

In Figure 1, because all the visual blocks are parallel under the DOM node *BODY*, we get many horizontal separators. In the first iteration, the separator with a *HR* in its position gets the maximal weight. In the second iteration, the separator below "THE RESKS DYGEST" is selected because of its Font pattern.

Step 3: Content Structure Construction

When the actual separators are detected, visual blocks on the same sides of all the separators are merged and represented as a node in the content structure. The *DoC* of each node is also calculated through similar methods as described in Step 1. After that, each node is checked whether it meets the granularity requirement. For every node that fails to meet the requirement, we go to Step 1 again to further construct the sub content structure within the node. If all the nodes meet the requirement, the iterative process is then stopped and the vision-based content structure for the whole page is obtained. The common requirement for *DoC* is that *DoC* > *PDoC*, if *PDoC* is pre-defined.

Go back to the example in Figure 1. In the first iteration, the actual separator found in Step 2 results two nodes *VB1-1* and *VB1-2*. *VB1-1* meets the *DoC* requirement and needs no further processing, while *VB1-2* goes through another iteration and gets several children. After several iterations, the final vision-based content structure of the page is constructed.

2.3 Discussion

The proposed VIPS algorithm takes advantage of visual cues to obtain the vision-based content structure of a web page. The algorithm successfully bridges the gap between the DOM structure and the semantic structure. The page is partitioned based on visual separators and structured as a hierarchy closely related to how a user will browse the page. Content related parts could be grouped together even if they are in different branches of the DOM tree.

VIPS is also very efficient. Since we trace down the DOM structure for visual block extraction and do not analyze every basic DOM node, the algorithm is totally top-down. Furthermore, the *PDoC* can be pre-defined, which brings significant flexibility to segmentation and greatly improve the performance. In our experiments, the average time to produce the content structure for a web page is 0.2 second. Since VIPS heavily relies on an analyzer to extract visual information, a majority of the time is spent on rendering and displaying the web page. As a consequence, VIPS is slower than those algorithms that only use text or DOM structure. If a light DOM parser without the displaying process is used, the performance of VIPS can be improved dramatically.

Although a complex page usually corresponds to a more complicated content structure, in VIPS, it is sometimes easier to partition these pages, since more visual cues can be detected and utilized. An example of complex pages is shown in [6], in which a formal definition of the vision-based content structure is presented and a more detailed description of the segmentation algorithm is illustrated.

Furthermore, those badly presented web pages are still posing a big challenge to VIPS. We are exploring how to employ a learning approach to obtain more robust rules of page segmentation.

3. APPLYING VIPS ON PSEUDO-RELEVANCE FEEDBACK

Since our VIPS algorithm can group semantically related content into a single segment, the term correlations within a segment will be much higher than those within other range of a whole web page. With the improved term correlations, high-quality expansion terms can be extracted from segments and then used to improve information retrieval performance. Therefore, we combine VIPS with the pseudo-relevance feedback algorithm according to the following steps:

Step 1: Initial Retrieval

An initial list of ranked web pages is obtained by using any traditional information retrieval methods.

Step 2: Page Segmentation

In this step, VIPS algorithm is applied to divide retrieved web pages into segments. After the vision-based content structure is obtained, all the leaf nodes are extracted as segments. Since it is very expensive to process all retrieved web pages, we only select a few (e.g. 80) top pages for segmentation. The candidate segment set is made up of these resulting segments.

Step 3: Segment Selection

This step aims to choose the most relevant segments from the candidate segment set. Some ranking methods (such as BM2500 [16]) are used to sort the candidate segments and the top (e.g. 20) segments are selected for expansion term selection in the next step.

Step 4: Expansion Term Selection

We use an approach similar to the traditional pseudo-relevance feedback algorithm to select expansion terms. The difference is that expansion terms are selected from the selected segments instead of the whole web pages. All terms except the original query terms are weighted according to the following term selection value:

$$TSV = w^{(1)} * r / R \tag{1}$$

where $w^{(1)}$ is the Robertson/Sparck Jones weight [16] (see (3)); R is the number of selected segments; and r is the number of segments which contain this term. In our experiments, the top 10 terms are selected to expand the original query.

Step 5: Final Retrieval

The term weights for the expanded query are set as follows:

- ✧ For original terms, new weight is $tf * 2$ where tf is its term frequency in the query;
- ✧ For expansion terms, new weight is $1-(n-1)/10$ if the current term ranks nth in TSV rank. Note that we assume a total of 10 terms are selected to expand the query.

The expanded query is used to retrieve the data set again for the final results.

4. EVALUATION

This section provides empirical evidences on how web page segmentation can be used to improve the performance of pseudo-relevance feedback in Web information retrieval. We compare our vision-based page segmentation method with simple DOM-based approach and full document search briefly described below.

- ✧ Our Vision-based approach (denoted as VIPS): The *PDoC* is set to 0.6. To reduce the effect of tiny blocks, blocks less than 10 words are removed. The top 80 pages returned by the initial retrieval phase are segmented to form the candidate segment set.
- ✧ Simple DOM-based approach (denoted as DOMPS): We iterate the DOM tree for some structural tags such as *TITLE, P, TABLE, UL* and *H1~H6*. If there are no more structural tags within the current structural tag, a block is constructed and identified by this tag. Free text between two tags is also treated as a special block. Similar to VIPS, tiny blocks less than 10 words are also removed, and the candidate segments are chosen from the top 80 pages returned by the initial retrieval phase.
- ✧ Traditional full document approach (denoted as FULLDOC): The traditional pseudo-relevance feedback based on the whole web page is implemented for a comparison purpose.

4.1 Test Data and Configuration

We choose Okapi [15] as the retrieval system and WT10g [2] in TREC-9 and TREC 2001 Web Tracks as the data set. WT10g contains 1.69 million pages and amounts to about 10G. We use the 50 queries from TREC 2001 Web Track as the query set and only the TOPIC field for retrieval.

We use BM2500 in Okapi for the weight function. It is of the form

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf\,(k_3+1)qtf}{(K+tf)(k_3+qtf)} \tag{2}$$

where Q is a query containing key terms T, tf is the frequency of occurrence of the term within a specific document, qtf is the frequency of the term within the topic from which Q was derived, and $w^{(1)}$ is the Robertson/Spark Jones weight of T in Q. It is calculated by

$$\log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \tag{3}$$

where N is the number of documents in the collection, n is the number of documents containing the term, R is the number of documents relevant to a specific topic, and r is the number of relevant documents containing the term. In (2), K is calculated by

$$k_1((1-b)+b \times dl / avdl) \tag{4}$$

where dl and $avdl$ denote the document length and the average document length measured in some suitable unit, such as word or a sequence of words. In our experiments, we set $k_1 = 1.2$, $k_3 = 1000$, $b = 0.75$, and $avdl = 61200$.

In our experiments, we do not use any stemming method. A word list containing 222 words [17] is used to filter out stop words. We only use single word and do not consider phrase information.

4.2 Experimental Results

To show the effect of segment selection, we alter the number of selected segments from 3 to 60 in the experiments. Table 1 and Figure 3 illustrate the comparison of performance. In Table 1, for each segmentation algorithm, the value in bold is the best performance achieved. In Figure 3, the horizontal axis is the number of documents or blocks, and the vertical axis shows the average precision. The baseline is 16.55% and is shown with a dotted line. Results from different web page segmentation algorithms are shown respectively.

As can be seen, the average retrieval precision can be improved after partitioning pages into blocks, no matter which segmentation algorithm is used. In the case of FULLDOC, the maximal average precision is 19.10% when the top 10 documents are used to expand the query. DOMPS obtains 19.67% when the top 50 blocks are used, a little better than FULLDOC. VIPS gets the best result 20.98% when the top 20 blocks are used and achieves 26.77% improvement.

Table 1. Performance comparison of query expansion using different page segmentation methods

Number of Segments	Baseline (%)	FULLDOC (%)	DOMPS (%)	VIPS (%)
3	16.55	17.56 (+6.10)	17.94 (+8.40)	18.01 (+8.82)
5		17.46 (+5.50)	18.15 (+9.67)	19.39 (+17.16)
10		**19.10 (+15.41)**	18.05 (+9.06)	19.92 (+20.36)
20		17.89 (+8.10)	19.24 (+16.25)	**20.98 (+26.77)**
30		17.40 (+5.14)	19.32 (+16.74)	19.68 (+18.91)
40		15.50 (-6.34)	19.57 (+18.25)	17.24 (+4.17)
50		13.82 (-16.50)	**19.67 (+18.85)**	16.63 (+0.48)
60		14.40 (-12.99)	18.58 (+12.27)	16.37 (-1.09)

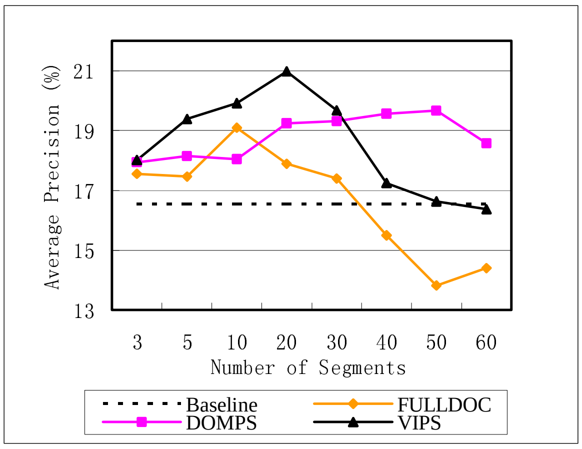


Figure 3. Performance comparison of query expansion using different page segmentation methods

Document based query expansion FULLDOC uses all the terms within the top documents for expansion. Since the baseline is very low, many of top ranked documents are actually irrelevant and there are many terms coming from irrelevant topics. These cause the retrieval performance relatively low although better than the baseline. For the same reason, the average precision drops quickly when more documents are used to select expansion terms. It becomes lower than the baseline after 30.

DOM based approach DOMPS does not obtain a significant improvement compared to FULLDOC, partly because the segmentation is too detailed. The average length of segments in DOM based approach is about 540 in byte. The segments are usually too short to cover complete information about a single semantic. In many cases, good expansion terms are within the previous or proceeding blocks, but are missed because those blocks are not ranked high enough to be selected in pseudo-relevance feedback.

Compared with DOMPS, our VIPS algorithm considers more visual information and is more likely to obtain a semantic partition of a web page. Therefore, more good expansion terms can be extracted and better performance can be achieved. As can be seen from Figure 3, VIPS performs the best when less than 20 blocks are used in query expansion. The reason is that the top blocks are usually most relevant and thus contain very good expansion terms. After 20 blocks, more noises is introduced and the performance drops quickly. Some other reasons for the quick drop lie in that many pages in our experiments are badly presented and that the block lengths vary remarkably.

From Figure 3 we can also see different performance peaks for the three approaches, which reflect different granularities of page segmentation. FULLDOC takes whole document for expansion term selection and has coarsest granularity, therefore arrives at its peak when only 10 documents are selected. DOMPS is the most detailed approach and gets the best performance when 50 blocks are used. VIPS gets its peak between the former two.

Table 2 provides the comparison of every recall level using the best result of each approach listed in Table 1. The VIPS approach performs the best in most recall levels.

Table 2. Comparison of precision at all recall levels using best results of different page segmentation methods

Recall (%)	Baseline (%)	FULLDOC (%)	DOMPS (%)	VIPS (%)
0	58.55	**62.04 (+5.96)**	58.87 (+0.54)	59.31 (+1.30)
10	37.09	40.59 (+9.44)	**41.42 (+11.67)**	40.16 (+8.28)
20	28.13	30.43 (+8.18)	32.73 (+16.35)	**34.33 (+22.04)**
30	21.35	24.84 (+16.35)	27.09 (+26.89)	**29.69 (+39.06)**
40	16.94	21.19 (+25.09)	22.61 (+33.47)	**23.54 (+38.96)**
50	14.33	17.60 (+22.82)	19.02 (+32.73)	**21.00 (+46.55)**
60	10.61	13.33 (+25.64)	14.26 (+34.40)	**15.90 (+49.86)**
70	7.66	9.87 (+28.85)	10.56 (+37.86)	**11.66 (+52.23)**
80	5.96	6.65 (+11.58)	6.34 (+6.38)	**8.59 (+44.13)**
90	3.97	3.85 (-3.02)	3.68 (-0.73)	**4.98 (+25.44)**
100	1.95	0.96 (-50.77)	0.97 (-50.26)	**1.99 (+2.05)**
Avg.	16.55	19.10 (+15.41)	19.67 (+18.85)	**20.89 (+26.77)**

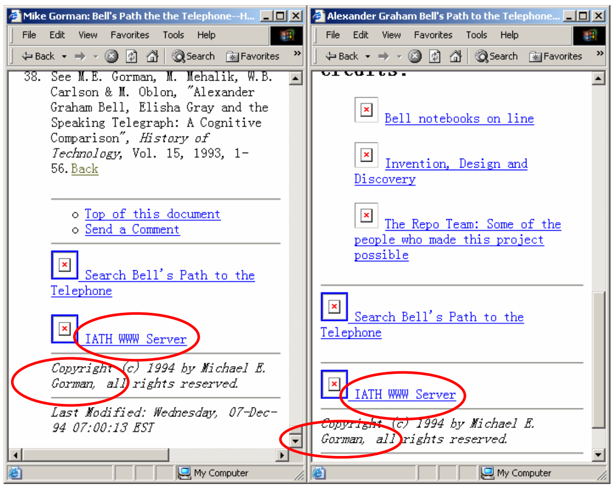
4.3 Case Studies

In the above experiments, we observe that the VIPS algorithm successfully overcome the problems of noise and multi-topics.

Let us take query #15 as an example to show how the impact of noise can be eliminated by page segmentation. Query #15 is "what about alexander graham bell?" that aims to retrieve relevant web pages about inventions by Alexander Graham Bell. The baseline precision of this query is 16.05%. Table 3 lists the expansion terms and precisions of this query for different methods.

Table 3. Expansion terms obtained for query #15

Methods	FULLDOC	DOMPS	VIPS
Expansion Terms	iath notebooks deaf invention gorman simulating 165 1876 inventors invent	audiotelegraph 1876 mehalik oblon prototype industriously technology tinkerer telephone spattered	invention deaf 1876 telephone divestiture edison telephones lineage 1885 inventors
Avg. Precision	14.13%	35.06%	50.89%


Mike Gorman: Bell's Path the the Telephone--H...

38. See M.E. Gorman, M. Mehalik, W.B. Carlson & M. Oblon, "Alexander Graham Bell, Elisha Gray and the Speaking Telegraph: A Cognitive Comparison", *History of Technology*, Vol. 15, 1993, 1-56. Back

- Top of this document
- Send a Comment

Search Bell's Path to the Telephone

IATH WWW Server

Copyright (c) 1994 by Michael E. Gorman, all rights reserved.

Last Modified: Wednesday, 07-Dec-94 07:00:13 EST

Alexander Graham Bell's Path to the Telephone...

Bell notebooks on line

Invention, Design and Discovery

The Repo Team: Some of the people who made this project possible

Search Bell's Path to the Telephone

IATH WWW Server

Copyright (c) 1994 by Michael E. Gorman, all rights reserved.

Figure 4. Example pages for query #15

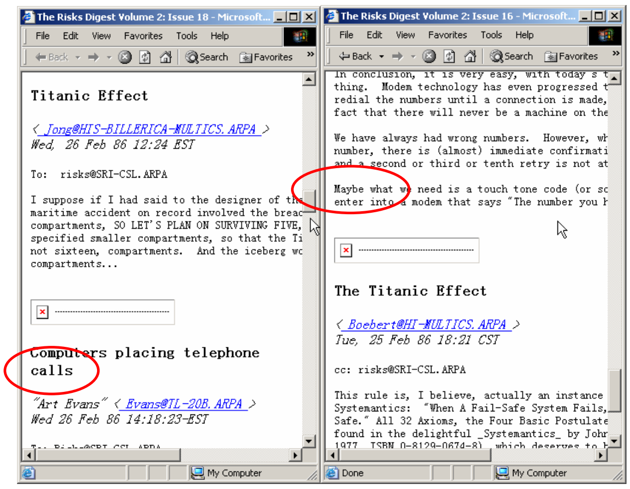
(DOCNO: WTX003-B35-115 and WTX004-B09-289)

There are two strange words "iath" and "gorman" that are selected as expansion terms by FULLDOC. When we investigate the top 10 pages (two examples are shown in Figure 4), we found that the word "iath" appears on the bottom of the page "IATH WWW Server" and "gorman" is the author of the book "Alexander Graham Bell's Path to the Telephone", so his name is on the bottom of the page as the copyright owner. Since these two words are not directly related to the original query, the final retrieval gets many unrelated documents containing these words, which decreases the search precision.

To show how the multi-topics problem is dealt with by page segmentation, we choose query #17 as an example. The query is "titanic what went wrong" and the baseline precision is 6.71%. Within the top ranked web pages returned by the initial retrieval, all of the first, second and ninth documents are from the website "The Risks Digest" and have a similar structure as that in Figure 1. Two of them are shown in Figure 5. Because there are so many different topics in a single page, terms about the reasons leading to the sinking of the Titanic are hard to extract from the page. Many irrelevant words such as "call" and "maybe" in other topics are selected, which hurts the final retrieval performance. Table 4 shows the comparison of the selected expansion terms and precisions.

Table 4. Expansion terms obtained for query #17

Methods	FULLDOC	DOMPS	VIPS
Expansion Terms	iceberg sri call maybe actually noticed difference picked theater reasons	britannic sinking rms iceberg hmhs passengers tragedy ballard sank seawifs	sinking unsinkable iceberg britannic rms passengers 1912 ship cyberflix ocean
Avg. Precision	16.20%	15.23%	30.43%


The Risks Digest Volume 2: Issue 18 - Microsoft...

Titanic Effect

< *Jong@HIS-BILLERICA-MULTICS.ARPA* >
Wed, 26 Feb 86 12:24 EST

To: risks@SRI-CSL.ARPA

I suppose if I had said to the designer of th
maritime accident on record involved the breac
compartments, SO LET'S PLAN ON SURVIVING FIVE,
specified smaller compartments, so that the Ti
not sixteen, compartments. And the iceberg wo
compartments...

Computers placing telephone calls

"Art Evans" < Evans@TL-20B.ARPA >
Wed 26 Feb 86 14:18:23-EST

The Risks Digest Volume 2: Issue 16 - Microsoft...

In conclusion, it is very easy, with today's t
thing. Modem technology has even progressed t
redial the numbers until a connection is made,
fact that there will never be a machine on the

We have always had wrong numbers. However, wh
number, there is (almost) immediate confirmati
and a second or third or tenth retry is not at

Maybe what we need is a touch tone code (or sc
enter into a modem that says "The number you h

The Titanic Effect

< *Boebert@HI-MULTICS.ARPA* >
Tue, 25 Feb 86 18:21 CST

cc: risks@SRI-CSL.ARPA

This rule is, I believe, actually an instance
Systemantics: "When A Fail-Safe System Fails,
Safe." All 32 Axioms, the Four Basic Postulate
found in the delightful _Systemantics_ by Johr

Figure 5. Example pages for query #17

(DOCNO: WTX011-B22-202 and WTX011-B22-199)

In summary, our experiments showed that page segmentation, if done properly, is an effective way to detect and filter out noisy and irrelevant information. It also enables better expansion terms selection for improving the retrieval performance. Among the different page segmentation methods, VIPS proved to produce more accurately web page segmentation at the semantic level and significantly outperformed the simple DOMPS.

5. CONCLUSION AND FUTURE WORK

Web information retrieval shows many differences compared to traditional text retrieval. Some traditional approaches which work well in traditional text retrieval, such as pseudo-relevance feedback, could not achieve the same level of performance in web information retrieval if applied directly.

We presented in this paper a novel web page segmentation algorithm called VIPS which constructs the content structure of a web page based on visual cues. This content structure better reflects the semantic structure of the web page. The algorithm is very efficient and scalable, and thus suitable for real time processing.

We applied VIPS to assist pseudo-relevance feedback in web information retrieval. The experimental results demonstrated that by partitioning a web page into semantically related units, better query expansion terms can be selected to improve the overall retrieval performance. We also compared our VIPS approach with DOM-based approach as well as full documents based approach. The results indicated that the best performance is achieved by our VIPS approach.

In the future, we are planning to apply VIPS to other web information retrieval methods, such as passage retrieval, and will compare the performance of VIPS with other traditional methods such as the fixed-length window method. We are also exploring other applications of VIPS, such as information extraction and topic distillation.

6. REFERENCES

[1] Baeza-Yates, R. and Ribeiro-Neto, B., Modern Information Retrieval, Addison Wesley Longman 1999.

[2] Bailey, P., Craswell, N., and Hawking, D., Engineering a multi-purpose test collection for Web retrieval experiments, Information Processing and Management, 2001.

[3] Buckley, C., Salton, G., and Allan, J., Automatic Retrieval with Locality Information Using Smart, In the First Text REtrieval Conference (TREC-1), National Institute of Standards and Technology, Gaithersburg, MD, 1992, pp. 59-72.

[4] Buttler, D., Liu, L., and Pu, C., A Fully Automated Object Extraction System for the World Wide Web, In International Conference on Distributed Computing Systems, 2001.

[5] Buyukkokten, O., Garcia-Molina, H., and Paepche, A., Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones, In Proceedings of the Conference on Human Factors in Computing Systems, CHI'01, 2001.

[6] Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y., Extracting Content Structure for Web Pages based on Visual Representation, In The Fifth Asia Pacific Web Conference (APWeb2003), 2003.

[7] Chakrabarti, S., Integrating the Document Object Model with hyperlinks for enhanced topic distillation and information extraction, In the 10th International World Wide Web Conference, 2001.

[8] Chen, J., Zhou, B., Shi, J., Zhang, H., and Wu, Q., Function-Based Object Model Towards Website Adaptation, In Proceedings of the 10th International World Wide Web Conference, 2001.

[9] Crivellari, F. and Melucci, M., Web Document Retrieval Using Passage Retrieval, Connectivity Information, and Automatic Link Weighting--TREC-9 Report, In The Ninth Text REtrieval Conference (TREC 9), 2000.

[10] Efthimiadis, N. E., Query Expansion, In Annual Review of Information Systems and Technology, Vol. 31, 1996, pp. 121-187.

[11] Embley, D. W., Jiang, Y., and Ng, Y.-K., Record-boundary discovery in Web documents, In Proceedings of the 1999 ACM SIGMOD international conference on Management of data, Philadelphia PA, 1999, pp. 467-478.

[12] Gu, X., Chen, J., Ma, W.-Y., and Chen, G., Visual Based Content Understanding towards Web Adaptation, In Second International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH2002), Spain, 2002, pp. 29-31.

[13] Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski, S., and Laakko, T., Two Approaches to Bringing Internet Services to WAP Devices, In Proceedings of 9th International World-Wide Web Conference, 2000, pp. 231-246.

[14] Newby, G., Information Space Based on HTML Structure, In The Ninth Text REtrieval Conference (TREC 9), 2000, pp. 601-610.

[15] Robertson, S. E., Overview of the okapi projects, Journal of Documentation, Vol. 53, No. 1, 1997, pp. 3-7.

[16] Robertson, S. E. and Sparck Jones, K., Relevance weighting of search terms, Journal of the American Society of Information Science, Vol. 27, No. May-June, 1976, pp. 129-146.

[17] Robertson, S. E. and Walker, S., Okapi/Keenbow at TREC-8, In the Eighth Text REtrieval Conference (TREC 8), 1999, pp. 151-162.

[18] Yang, Y. and Zhang, H., HTML Page Analysis Based on Visual Cues, In 6th International Conference on Document Analysis and Recognition (ICDAR 2001), Seattle, Washington, USA, 2001.