Block-based Web Search

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

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

[1]Tsinghua University
Beijing, China
cai_deng@yahoo.com

[2]Institute for Computer Science
University of Munich
yushipeng@yahoo.com

ABSTRACT

Multiple-topic and varying-length of web pages are two negative factors significantly affecting the performance of web search. In this paper, we explore the use of page segmentation algorithms to partition web pages into blocks and investigate how to take advantage of block-level evidence to improve retrieval performance in the web context. Because of the special characteristics of web pages, different page segmentation method will have different impact on web search performance. We compare four types of methods, including fixed-length page segmentation, DOM-based page segmentation, vision-based page segmentation, and a combined method which integrates both semantic and fixed-length properties. Experiments on block-level query expansion and retrieval are performed. Among the four approaches, the combined method achieves the best performance for web search. Our experimental results also show that such a semantic partitioning of web pages effectively deals with the problem of multiple drifting topics and mixed lengths, and thus has great potential to boost up the performance of current web search engines.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Human Factors

Keywords

Web information retrieval, page segmentation, passage retrieval, query expansion

1. INTRODUCTION

Passage retrieval is a research topic with long history in the IR community which addresses the shortcomings of whole-document ranking. Previous work reveals that it is sometimes beneficial to apply retrieval algorithms to portions of a document, particularly when documents contain multiple drifting subjects or have varying lengths [5][10][18].

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

The Web today contains documents that are highly volatile, distributed and heterogeneous. The content of a web page is usually much more diverse compared with traditional plain text document and encompasses multiple regions with unrelated topics. Moreover, for the purpose of browsing and publication, non-content materials, such as navigation bars, decoration stuffs, interaction forms, copyrights, and contact information, are usually embedded in web pages. Instead of treating a whole web page as a unit of retrieval, we argue that the characteristics of web pages make passage a more effective mechanism for information retrieval.

The major shortcoming of treating a web page as a single semantic unit is that it does not consider multiple topics in a page. For example, if the query terms scatter at various regions with different topics, it could cause low retrieval precision. It can be argued that a web page with a region of high density of matched terms is likely to be more relevant than a web page with matched terms distributed across the entire page even if it has higher overall similarity. On the other hand, a highly relevant region in a web page may be obscured because of low overall relevance of that page.

In addition, correlations among terms in a web page may be inappropriately calculated if the web page contains multiple unrelated topics, which, in turn, is a negative factor for query expansion. Take pseudo-relevance feedback as an example, if an advertisement is embedded in a top-ranked web page at the first retrieval, then some terms from the advertisement may be selected as expansion terms. Once these irrelevant terms are used to expand the query for the second retrieval, it may decrease the retrieval performance. Therefore, it is necessary to segment a web page into semantically independent units (i.e. web page blocks) so that noisy information can be filtered out and multiple topics can be distinguished.

It is well known that in document retrieval the similarity measure is very sensitive to document length, and some measures, such as the Cosine measure, tend to favor short documents, resulting in a biased result. To understand how the length of web page is varied, we conducted a statistical study on TREC's WT10g [2] and GOV [1] data sets, compared with traditional document sets TREC-24 (TREC disks 2&4) and TREC-45 (TREC disks 4&5) [11]. As shown in Table 1, the two web data sets show more difference between average length and medium length, and thus suffer more length variance. To deal with this problem, some length normalization methods for plain texts have been proposed, but finding a uniform solution for a wide range of document collections is still a difficult problem. Previous work showed that partitioning a document into passages, especially fixed-length passages, can reduce the difficulty of document length normalization [5][10]. But to the best of our knowledge, there is no thorough comparisons reported on the web data set.

Table 1. Comparison of free-text and web document sets

	TREC-24	TREC-45	**WT10g**	**GOV**
Number of doc	524,929	556,077	**1,692,096**	**1,247,753**
Text size (Mb)	2,059	2,134	**10,190**	**18,100**
Median length (Kb)	2.5	2.5	**3.3**	**7.5**
Average length (Kb)	4.0	3.9	**6.3**	**15.2**

It is clear that web pages suffer from the same, if not worse, problems of multiple topics and varying length as plain text documents. In this paper, we investigate how to take advantage of block-level evidence to improve information retrieval on the web. As the central part of this work relies on a good *web page segmentation* scheme, we first conduct a thorough comparison on four page segmentation approaches for improving web information retrieval. Experiments show that, similar to the case of plain-text retrieval, partitioning the web pages into smaller units will significantly improve the retrieval performance. Furthermore, unlike fixed-window's great importance to plain-text retrieval, semantic partitioning can be easier and more accurate to implement on the web context and plays a more crucial role for web information retrieval. Among all the page segmentation algorithms, the best performance is achieved by a combined algorithm which integrates both semantic and fixed-length methods.

The rest of the paper is organized as follows. Section 2 discusses the particular characteristics of passage extraction for web pages and some previous works. Four types of web page segmentation approaches are introduced in Section 3. Experiments of applying these page segmentation methods on block-level retrieval and query expansion are presented in Section 4. Section 5 summarizes our contributions and concludes the paper.

2. WEB PAGE SEGMENTATION

In traditional passage retrieval, passages can be categorized into three classes: discourse, semantic, and window. Discourse passages rely on the logical structure of the documents marked by punctuation, such as sentences, paragraphs and sections [5][18][20]. Semantic passages are obtained by partitioning a document into topics or sub-topics according to its semantic structure [9][16][19]. A third type of passages, fixed-length passages or windows, are defined to contain fixed number of words [5][24][11].

While directly adopting these passage definitions for partitioning web pages is feasible, there exist some new characteristics in web pages which can be utilized. We describe each of them below:

- *Two-Dimension Logical Structure* – Different from plain-text documents, web pages have a 2-D view and a more sophisticated internal content structure. Each region of a web page could have relationships with regions from up to four directions and contain or be contained in some other regions. A content structure in semantic level exists for most pages and can be used to enhance retrieval.
- *Visual Layout Presentation* – To facilitate browsing and attract attention, web pages usually contain much visual information in the tags and properties in HTML [22]. Typical visual hints include lines, blank areas, colors, pictures, fonts, etc. Visual cues are very helpful to detect the semantic regions in web pages.

Due to the 2-D logical structure, web pages could be partitioned in a 2-D style. Therefore, instead of using "passage", we prefer to use *block* to denote a region of web pages. A block is assumed to have a rectangle shape and is a closely packed region in the original page. Accordingly, the process of partitioning web pages into blocks is called *web page segmentation*.

There have been some research works on web page segmentation and its applications. In [12][15][14], traditional passages are used to partition web pages, but the results are not encouraging, which verifies that traditional passages might not appropriate for web context, and that we need to consider more characteristics of web documents.

Some approaches rely on the DOM (Document Object Model, see http://www.w3.org/DOM/), since DOM provides a hierarchical structure for every web page. Some useful tags or tag types are used to identify blocks [13][21], including <P>, <TABLE>, <UL>, <H1>~<H6>, etc. Some other works also consider extra information such as content [8] and link [6]. However, all these methods are not targeting on web information retrieval and thus are difficult to evaluate and compare. Some simple experiments have been performed on web information retrieval [7] but little improvement is obtained, partly because DOM is still a kind of linear structure and usually unable to represent the semantic structure of a page. From this perspective, DOM based blocks are, in some sense, similar to traditional discourse passages.

To take full advantage of new characteristics of web pages, we have proposed a more effective page segmentation technique called VIPS (VIsion-based Page Segmentation) in [3][4], in which various visual cues are taken into account to achieve a more accurate content structure on the semantic level. We also showed that this method can greatly improve the performance of pseudo-relevance feedback [23]. However, the blocks obtained from VIPS still have the varying length problem and suffer from lack of normalization factor. More importantly, it remains unclear whether the method would work on passage retrieval and no comparison is provided between this method and traditional passage retrieval methods such as windows, which can be naturally applied to web documents.

To deal with the shortcomings of VIPS, in this paper we introduce a combined algorithm which takes advantage of both visual layout and length normalization. A web page will first be passed to VIPS for segmentation, and then to a normalization procedure. Therefore, this algorithm can deal with all the problems we have mentioned in Section 1.

We further compare four kinds of web page segmentation methods in this paper: fixed-length page segmentation (FixedPS), DOM-based page segmentation (DomPS), vision-based page segmentation (VIPS), and the combined method CombPS. Unlike in [23], experiments on both block-level query expansion and retrieval are conducted based on all of these methods, using two different web document sets. The experimental results verify that page segmentation is very effective in dealing with the multiple-topic and varying length problems of web pages, and therefore can significantly improve the overall retrieval performance. Among all these page segmentation methods, the combined method achieves the best performance in all the experiments.

3. THE FOUR METHODS

In this section, we describe the four web page segmentation methods and compare them from theoretical prospective. A natural correspondence between these page segmentation methods and traditional passage retrieval methods is shown in Table 2.

Table 2. Correspondence between page segmentation methods and traditional passage retrieval methods

Web Page Segmentation	FixedPS	DomPS	VIPS	CombPS
Passage Retrieval	Window	Discourse	Semantic	Semantic Window

3.1 Fixed-length Page Segmentation (FixedPS)

In traditional text retrieval, fixed-length passages, or windows, are used to overcome the difficulty of length normalization. A fixed-length passage contains fixed number of continuous words. An overlapped window approach is proposed by Callan [5], in which the first window in one document starts at the first occurrence of a query term, and subsequent windows half-overlap preceding ones.

For web documents, fixed-length page segmentation is identical to traditional window approach except that all the HTML tags and attributes are removed. The length of window is the only parameter and is suggested to be 200 or 250 from past experience [5].

Despite its simplicity, fixed-length segmentation is very robust and effective for improving performance, particularly for collections with long or mixed-length documents [5][11]. The main shortcoming of the fixed-length method is that no semantic information is taken into account in the segmentation process.

3.2 DOM-based Page Segmentation (DomPS)

DOM provides each web page with a fine-grained structure, which illustrates not only the content but also the presentation of the page. In general, similar to discourse passages, the blocks produced by DOM-based methods tend to partition pages based on their pre-defined syntactic structure, i.e., the HTML tags.

There are some approaches that take into account the problem of page segmentation, but there is no consistent way to do it and, to the best of our knowledge, few works are done on applying DOM-based page segmentation methods on web information retrieval. Some simple experiments are performed in [7], where sub-trees tagged with <TITLE>, <P>, <H1>~<H3> and <META> are treated as blocks, but the results are not encouraging. The reasons may lie in the following three aspects. First, DOM is still a linear structure, so visually adjacent blocks may be far from each other in the structure and departed wrongly. Secondly, tags such as <TABLE> and <P> are used not only for content presentation but also for layout structuring. It is therefore difficult to obtain the appropriate segmentation granularity. Thirdly, in many cases DOM prefers more on presentation to content and therefore not accurate enough to discriminate different semantic blocks in a web page.

3.3 Vision-based Page Segmentation (VIPS)

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, such as lines, blanks, images, colors, etc [22]. For the sake of easy browsing and understanding, a closely packed block within the web page is much likely about a single semantic.

We have previously proposed a vision-based page segmentation method called VIPS in [4]. Similar to semantic passages, the blocks obtained by VIPS are based on the semantic structure of web pages. Traditional semantic passages are obtained based on content analysis which is very slow, difficult and inaccurate. VIPS discards content analysis and produce blocks based on the visual cues of web pages. This method simulates how a user understands web layout structure based on his or her visual perception. The DOM structure and visual information are used iteratively for visual block extraction, visual separator detection and content structure construction. Finally a *vision-based content structure* can be extracted. Since the method is totally top-down and the *permitted degree of coherence* can be pre-defined, the whole page segmentation procedure is efficient, flexible and more accurate from semantic perspective.

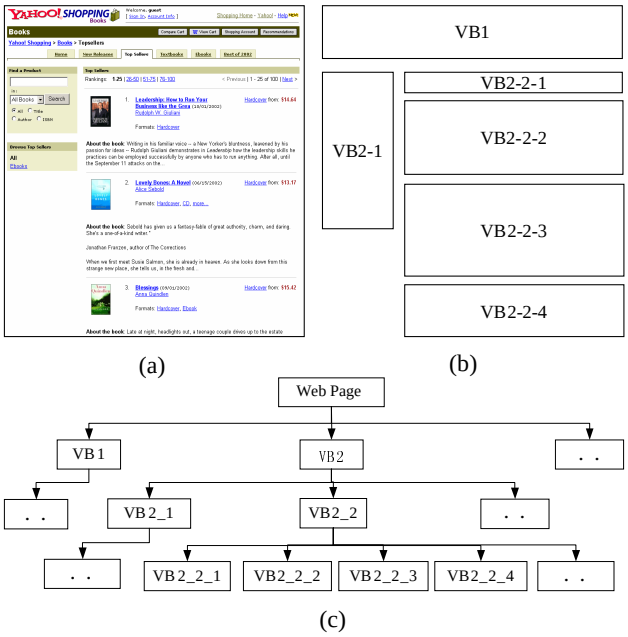


Figure 1. Vision-based content structure for the sample page

In Figure 1, the vision-based content structure of a sample page is illustrated. Visual blocks are detected as shown in Figure 1(b) and the content structure is shown in Figure 1(c). It is an approximate reflection of the semantic structure of the page.

In VIPS method, a visual block is actually an aggregation of some DOM nodes. Unlike DOM-based page segmentation, a visual block can contain DOM nodes from different branches in the DOM structure with different granularities. Structural tags such as <TABLE> and <P> can be divided appropriately with the help of visual information, and wrong presentation of DOM structure can be reorganized to a proper form. Therefore, VIPS can achieve a better content structure for the original web page.

3.4 A Combined Approach (CombPS)

Although VIPS can distinguish multiple topics in web pages, it does not consider the document length normalization problem. We have performed a statistical experiment on 50,000 pages retrieved from the WT10g dataset given 50 queries of TREC 2001. By using VIPS, we obtained totally 602,029 blocks. Figure 2 illustrates the block length distribution.

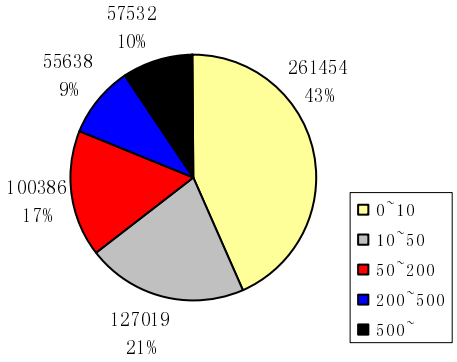


Figure 2. The distribution of block length after using VIPS to segment 50,000 pages chosen from the WT10g data set

As can be seen from this figure, the distribution of block length is very diverse. More than 40% of the blocks are only less than 10 words, and 10% blocks are larger than 500 words. Thus the varying length problem still exists even if we perform retrieval on block level.

Since fixed-length windows show great consistence on dealing with the varying length problem, we propose a combined page segmentation approach called CombPS which tries to take advantage of both visual information and fixed length. The CombPS method is processed as the following two steps:

Step 1. Vision-based Page Segmentation

The VIPS method described in Section 3.3 is used in this step. After the vision-based content structure is obtained, all the leaf visual blocks are taken as the input to the next step for block extraction.

Step 2. Fixed-length Block Extraction

For each visual block obtained in the previous step, overlapped windows are used to divide the block into smaller units. The first window begins from the first word of the visual block, and subsequent windows half-overlap preceding ones till the end of the block. For visual blocks that are smaller than the pre-defined length of the window, they are directly outputted as final blocks without further partition.

Upon this strategy, large visual blocks are departed into smaller ones and thus greatly reduce the impact of varying length. Compared with fixed-length approach FixedPS, CombPS utilizes semantic information in partitioning and makes page segmentation insensitive to queries. By allowing small semantic blocks to directly be parts of segmentation results, CombPS intuitively obtains a more diverse and "correct" segmentation result set.

4. WEB INFORMATION RETRIEVAL USING PAGE SEGMENTATION

In this section, we reported the experimental results of using different page segmentation methods on block-level retrieval and query expansion, respectively.

4.1 Methodology

The following four page segmentation methods are evaluated in our experiments. No specific tunings are applied to these methods.

- Fixed-length approach (FixedPS) - We use a similar approach as Callan's [5]. The window length is set to be 200 words.
- DOM-based approach (DomPS) - We iterate the DOM tree for some structural tags <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.
- Vision-based approach (VIPS) - The *permitted degree of coherence* is also set to 0.6. After the segmentation process, all the leaf nodes are extracted as visual blocks.
- The combined approach (CombPS) - This is the method described in Section 3.4. In the first step, the parameters are set to the same as VIPS; in the second step, the window length is set to be 200 words.

A full document approach (FullDoc) is also implemented for comparison purpose, in which no segmentation is performed and pages are treated as undivided units.

All of these page segmentation methods are evaluated on the following two important techniques of information retrieval.

Block Retrieval – Similar to passage retrieval, block retrieval performs the retrieval task at the block level and aims to adjust the rank of documents with the blocks they contain. Through this experiment, our main purpose is to verify whether page segmentation techniques are helpful to deal with both the length normalization and multiple-topic problems.

Query Expansion – For query expansion, expanded terms are extracted from relevant blocks, not the whole web pages. For this experiment, we aim to testify whether page segmentation can benefit the selection of query terms through increasing term correlations within a block, and thus improve the final performance.

4.2 Experiment Setup and Pre-processing

Our experiments are based on the Web Tracks of TREC 2001 and TREC 2002. The data set for TREC 2001 is "WT10g" which was crawled in 1997, and for TREC 2002 is ".GOV" which contains pages of 2002. We evaluated web page segmentation on both data sets using both query sets. Each query set contains 50 queries and only the <title> field is used for retrieval.

We choose Okapi [17] as the retrieval system and use BM2500 for the weight function. It is of the form

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

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 fre-

quency of the term within the topic from which Q was derived, and $w^{(1)}$ is the Robertson/Sparck Jones weight of T in Q. It is calculated by

$$\log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)}, \quad (2)$$

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 (1), K is calculated by

$$k_1((1-b)+b\times dl/avdl), \quad (3)$$

where dl and $avdl$ denote the document length and the average document length. To achieve the best baseline, we tune the parameters in our experiments and set k_3 = 1000, b = 0.25 for both data sets, but set k_1 = 0.5 for TREC 2001 and k_1 = 2.5 for TREC 2002, respectively.

A word list containing 222 words is used to filter out stop words. We do not use any stemming method and phrase information in our experiments, since our basic ideas are not related to these extra techniques and should also work without them.

In our experiments, the precision at 10 (P@10) is the main evaluation metric, and we also evaluate the average precision (AvP) for TREC 2001 since the Web Track in TREC 2001 is more on ad-hoc retrieval and is indeed evaluated by AvP. After the preprocessing, we get the retrieval baseline of 0.312 (AvP 0.1703) for TREC 2001 and 0.2286 for TREC 2002.

4.3 Experiments on Block Retrieval

The block retrieval experiments are conducted according to the following steps:

Step 1. Initial Retrieval

An initial list of ranked web pages is obtained by using the Okapi system. The document rank obtained in this step is called *DR*.

Step 2. Page Segmentation

A page segmentation method is applied to partition the retrieved pages into blocks. All of the extracted blocks form a block set.

Step 3. Block Retrieval

This step is similar to Step 1, except that documents are replaced by blocks. The same queries are used to get a block rank *BR*.

After obtaining the block rank, pages can be re-ranked based on the single best-ranked block within each page, though we can also consider several top blocks of each page to re-rank the page. Besides this simple approach, a combined rank is also presented in our experiments like in [5], in which the rank of each web page d is determined by $\alpha \cdot rank_{DR}(d) + (1-\alpha)\cdot rank_{BR}(d)$.

Table 3 shows the experimental results on block retrieval using different page segmentation methods. FullDoc is not listed here since it will always get the baseline. The third column shows the results of using single-best block rank, and the last column shows the results of combining block rank and document rank, with α being optimal for each specific method. The dependency between P@10 and α is illustrated in Figure 3, in which all the curves converge to the baseline when α = 1.

As can be seen from Table 3, if only the best block from each document is used to rank pages, DomPS performs the worst and FixedPS a little bit better, both of which are worse than the baseline for both data sets. VIPS is slightly better than baseline in TREC 2001 but fails to exceed baseline in TREC 2002, though it is the best among all the methods. CombPS wins TREC 2001, but is worse than VIPS in TREC 2002. For TREC 2002, no method can outperform the baseline.

When block rank is combined with the original document rank, the performance of all these four methods increases significantly and is better than the baseline. This shows the effect of rank combination, similar to traditional passage retrieval [5]. DomPS is still the worst, and FixedPS is slightly better. VIPS and CombPS are still better than the former two and show similar comparison characteristics to the non-combining situations, except that result of CombPS (0.2379) is now much closer to that of VIPS (0.2408) in TREC 2002.

Furthermore, from Figure 4 it can be seen that the winner for either data set shows a consistent improvement compared to the other methods, and thus does not win by chance. For TREC 2001 CombPS gets better performance almost in every combination, and for TREC 2002 CombPS shares rather similar trends as VIPS when α exceeds 0.4.

Table 3. P@10 Comparison on block retrieval

Page Segmentation	Baseline	*BR* only	*BR* + *DR* best
DomPS	0.312	0.252	0.322
FixedPS		0.304	0.326
VIPS		0.316	0.328
CombPS		**0.326**	**0.338**

(a) P@10 comparison for TREC 2001

Page Segmentation	Baseline	*BR* only	*BR* + *DR* best
DomPS	0.2286	0.1571	0.2286
FixedPS		0.1776	0.2317
VIPS		**0.2163**	**0.2408**
CombPS		0.1939	0.2379

(b) P@10 comparison for TREC 2002

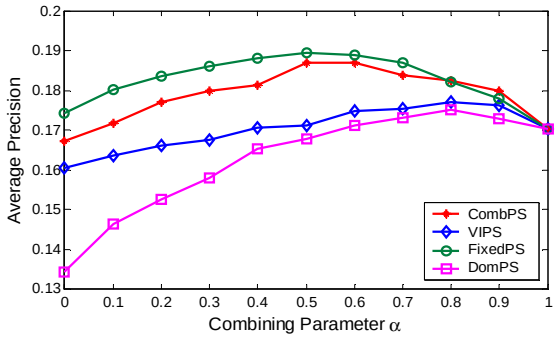
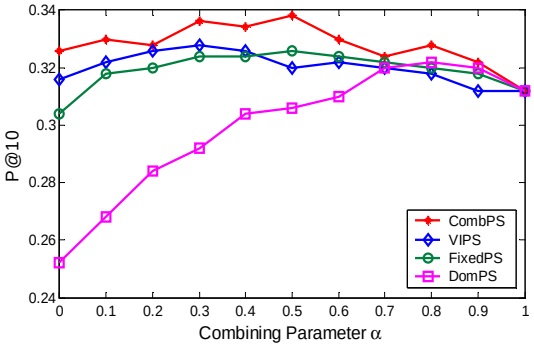
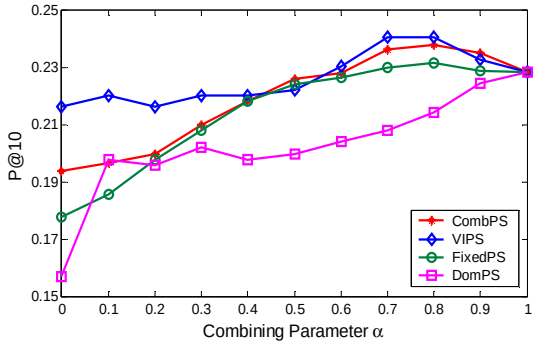


Figure 3. Comparisons of AvP with respect to combining parameter α for block retrieval on TREC 2001

To obtain a thorough comparison, we also evaluate all the methods by AvP for TREC 2001, as illustrated in Figure 3. FixedPS outperforms all the others in this situation and it is the only better-than-baseline method when no combination is utilized (i.e. α = 0).



(a) P@10 with different combining parameter on TREC 2001



(b) P@10 with different combining parameter on TREC 2002

Figure 4. Comparisons of web page segmentation methods on block retrieval. The x-axis is the combining parameter *α*, and the y-axis is the P@10 value. All the curves converge to the baseline when *α* = 1.

CombPS also shows very good performance and similar trend as FixedPS, but VIPS is much worse at this time. DomPS still performs the worst in all the combinations.

For a summarization for block retrieval, DomPS is always the worst and most unstable method, partly because the produced blocks are too detailed and usually can not be mapped to a single semantic part within the pages. FixedPS shows very good performance evaluated in AvP, which confirms the results of previous work on passage retrieval and testifies that varying-length is still an important factor to affect web information retrieval. However, FixedPS gives way to VIPS and CombPS when P@10 is the main concern, partly because it lacks semantic partition and fails to recognize best semantic blocks. VIPS is very good for both data sets in P@10, which means semantic partition is of great importance to web context, especially to newly crawled web pages (e.g., TREC 2002). But the inability to deal with varying length problem results a poor performance for VIPS when evaluated in AvP. Therefore, FixedPS and VIPS have different advantages and thus should be selected for different purposes. For web context, however, P@10 is more useful, which means semantic partition plays a more crucial role. By combining VIPS and FixedPS, CombPS aims to find a tradeoff between these two and therefore gets very good and stable performance. Whichever evaluation metric is used, CombPS is the best or very close to the best method. This shows that a combination of semantic structure and length normalization is the best choice for block retrieval.

4.4 Experiments on Query Expansion

In the experiments of query expansion, the first three steps are all the same as those of block retrieval. After block ranks are obtained, the following 4th and 5th steps are executed:

Step 4. Expansion Term Selection

Top-ranked blocks are used for expansion term selection. Expansion terms are selected in a way similar to the traditional pseudo-relevance feedback algorithm. All terms except the original query terms in the selected blocks are weighted according to the following term selection value *TSV*:

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

where $w^{(1)}$ is the same element described in (1), R is the number of selected blocks, and r is the number of blocks which contain this term. In our experiments, top 10 terms are selected to expand the original query.

Step 5. Final Retrieval

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

- For original terms, new weight is *tf* * 3 where *tf* is its term frequency in the query;
- For each expansion term, its weight is set to $1-(n-1)/m$, where n is the *TSV* rank value of this term, and m is the number of expansion terms, i.e., 10 in our experiments.

Then the expanded query is used to retrieve the collection again to get the final results.

We performed each web page segmentation method and chose top-ranked blocks (documents for FullDoc) to do query expansion. Figure 5 illustrates the P@10 values given different number of blocks (documents in FullDoc), and in Table 4, the P@10 value for each segmentation method is the best performance seen from Figure 5. Figure 6 also shows the same comparison for TREC 2001 by using average precision as the evaluation metric.

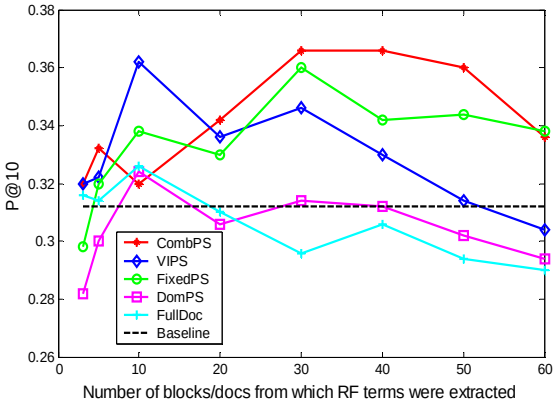
Table 4. P@10 comparison on query expansion

Page Segmentation	Baseline	Query Expansion (best)	
		P@10	Improvement
FullDoc	0.312	0.326	4.5%
DomPS		0.324	3.8%
FixedPS		0.36	15.4%
VIPS		0.362	16.0%
CombPS		**0.366**	**17.3%**

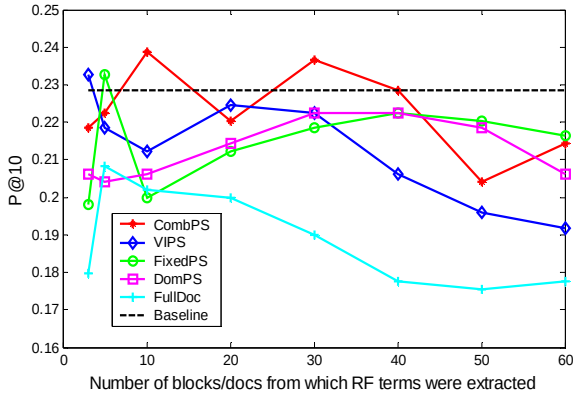
(a) P@10 comparison for TREC 2001

Page Segmentation	Baseline	Query Expansion (best)	
		P@10	Improvement
FullDoc	0.2286	0.2082	- 8.9%
DomPS		0.2224	- 2.7%
FixedPS		0.2327	1.8%
VIPS		0.2327	1.8%
CombPS		**0.2388**	**4.5%**

(b) P@10 comparison for TREC 2002



(a) P@10 with different number of blocks/docs on TREC 2001



(b) P@10 with different number of blocks/docs on TREC 2002

Figure 5. Comparisons of web page segmentation methods on query expansion. The x-axis is the number of blocks/docs from which RF terms were extracted, and the y-axis is the P@10 value. The baseline is shown with a dashed line.

From the experimental results, a general conclusion can be made that partitioning pages into blocks can improve the performance of query expansion, regardless of which page segmentation method is used. Furthermore, "good" segmentation method can improve the performance significantly and stably. Among all the page segmentation methods, FullDoc does nothing and thus may get good results (in TREC 2001) or bad results (in TREC 2002), but FixedPS, VIPS and CombPS can always get better results. DomPS is still unstable and sometimes even worse than the baseline. The performance of VIPS and FixedPS is similar, except that VIPS shows better performance in AvP, and that normally they achieve the peak at different number of blocks. CombPS, on the other hand, is always the best method and could achieve at most 17.3% improvement in P@10 and 28.5% in AvP.

We also performed various t-tests to check whether all these improvements are statistical significant. For TREC 2001, if compared with baseline, CombPS, VIPS, FixedPS are all significant (p-value is 0.0236, 0.0245 and 0.0466, respectively). FullDoc and DomPS, however, fail to pass the t-test (p-value is 0.156 and 0.291). This means "good" page segmentation methods can significantly improve the performance over the baseline and FullDoc. For TREC 2002, however, no methods show significant improvement over the baseline. If compared to FullDoc, only CombPS shows significant improvement (p-value is 0.048).

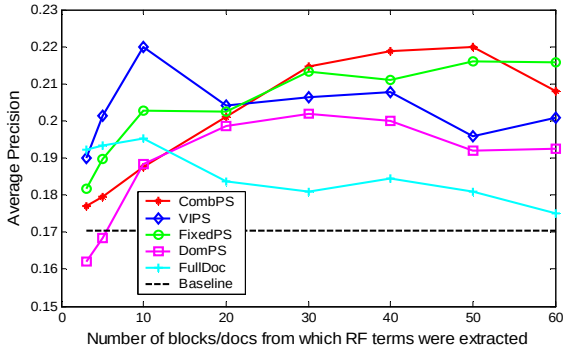


Figure 6. Comparisons of AvP versus number of blocks/docs for query expansion on TREC 2001

Since TREC 2002 aims for topic distillation, it seems that query expansion makes little improvement over the baseline. Although CombPS wins over other methods, it fails to show significant improvement.

We begin from FullDoc for a thorough comparison for query expansion. Since the baseline is very low, many of top ranked documents are actually irrelevant and there are many terms coming from irrelevant topics. Thus by using all the terms within top documents for expansion, FullDoc could only obtain a relatively low and insignificant result in all the experiments.

DomPS fails to obtain a significant improvement over the baseline and FullDoc, partly because the segmentation is too detailed. In our experiments, the average length in DomPS is only 540 in byte. After partitioning, although each block represents some information, it usually does not provide complete information about a single semantic, and thus does not contain good expansion terms.

Compared with DomPS, VIPS considers more visual information and is more likely to obtain a semantic partition of a web page. As seen from Figure 5 and Figure 6, VIPS tends to reach its best performance at a small number of blocks, which means top blocks usually have very good quality and thus can provide good expansion terms. We also notice that, for those "badly" presented web pages, VIPS usually fails to partition them into semantic blocks and thus expansion terms are likely to be irrelevant. Also, some relevant long blocks produced by VIPS are ranked low since our similarity measure tends to favor short documents.

FixedPS also achieves rather good performance. Since no structural and semantic information is considered in this method, in some cases it can deal with those "badly" presented pages. Since almost all blocks share the same length, there are no priorities for short blocks. As windows are overlapped, more blocks are likely to be extracted from a long document than VIPS, and thus FixedPS shows great steadiness when number of blocks increases. One problem of this approach is that no semantic information is considered. A window may cover contents from different semantic regions, and thus noisy terms are likely to be introduced.

Finally, CombPS shows better performance than both VIPS and FixedPS. Since blocks are partitioned based on semantic and vari-

ety of block length is relatively small, the shortcomings of VIPS and FixedPS are, to some extent, overcome.

For a brief summarization, semantic partition shows great importance for query expansion, and CombPS shows best performance.

5. CONCLUSION

In this paper we explored how to use web page segmentation to enhance web information retrieval and compared four methods extensively – namely fixed-length page segmentation, DOM-based page segmentation, vision-based page segmentation (VIPS), and a combined method which integrates both the vision-based and fixed-length properties. We evaluated the effectiveness of these page segmentations for block-level query expansion and retrieval, and verified that page segmentation can significantly improve the retrieval performance by dealing with the multiple-topic and mixed-length problems of web pages. Unlike fixed-window's great importance to plain-text retrieval, such a semantic partition is more important to the web context. By integrating semantic and fixed-length properties, we could deal with both problems and achieved the best performance. We believe such a block-level analysis of web pages will have the opportunity to significantly enhance the performance of existing commercial search engines. We plan to apply this technique to a data set close to the web scale in the future.

6. REFERENCES

[1] The .GOV test collection. TREC Web Tracks homepage. http://es.cmis.csiro.au/TRECWeb/.

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

[3] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, Extracting content structure for web pages based on visual representation, *Proc.5th Asia Pacific Web Conference*, Xi'an China, 2003.

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

[5] Callan, J. P., Passage-Level Evidence in Document Retrieval, In Proceedings of *the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, 1994, pp. 302-310.

[6] Chakrabarti, S., Joshi, M., and Tawde, V., Enhanced topic distillation using text, markup tags, and hyperlinks, In Proceedings of *the 24th annual international ACM SIGIR conference on Research and development in information retrieval* , ACM Press, 2001, pp. 208-216.

[7] 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.

[8] 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.

[9] Hearst, M. A., Multi-Paragraph Segmentation of Expository Text, In Proceedings of *the 32nd Annual Meeting of the Association for Computational Linguistics*, New Mexico State University, Las Cruces, New Mexico, 1994, pp. 9-16.

[10] Kaszkiel, M. and Zobel, J., Passage Retrieval Revisited, In Proceedings of *the 20th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1997, pp. 178-185.

[11] Kaszkiel, M. and Zobel, J., Effective Ranking with Arbitrary Passages, *Journal of the American Society for Information Science*, Vol. 52, No. 4, 2001, pp. 344-364.

[12] Kwok, K. L., Grunfeld, L., Dinstl, N., and Chan, M., TREC-9 Cross Language, Web and Question-Answering Track Experiments using PIRCS, In *The Ninth Text REtrieval Conference (TREC 9)*, 2000, pp. 419-427.

[13] Lin, S.-H. and Ho, J.-M., Discovering Informative Content Blocks from Web Documents, In Proceedings of *ACM SIGKDD'02*, 2002.

[14] Liu, S., Yu, C. and Wu, W., UIC at TREC-2002: Web Track. In *The Eleventh Text REtrieval Conference (TREC 2002)*, 2002

[15] Namba, I., Fujitsu Laboratories TREC-9 Report, In *The Ninth Text REtrieval Conference (TREC 9)*, 2000, pp. 203-208.

[16] Ponte, J. M. and Croft, W. B., Text Segmentation by Topic, In Proceedings of *the 1st European Conference on Research and Advanced Technology for Digital Libraries*, 1997

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

[18] Salton, G., Allan, J., and Buckley, C., Approaches to passage retrieval in full text information systems, In Proceedings of the *16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, Pennsylvania, USA, 1993, pp. 49-58.

[19] Salton, G., Singhal, A., Buckley, C., and Mitra, M., Automatic Text Decomposition Using Text Segments and Text Themes, In Proceedings of *the Seventh ACM Conference on Hypertext (Hypertext'96)*, ACM Press, New York, 1996

[20] Wilkinson, R., Effective Retrieval of Structured Documents, In Proceedings of *the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Dublin, 1994, pp. 311-317.

[21] Wong, W. and Fu, A. W., Finding Structure and Characteristics of Web Documents for Classification, In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, Dallas, TX., USA, 2000.

[22] 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.

[23] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma, Improving pseudo-relevance feedback in web information retrieval using web page segmentation, *Proc. 12th World Wide Web Conference*, Budapest, Hungary, 2003.

[24] Zobel, J., Moffat, A., Wilkinson, R., and Sacks-Davis, R., Efficient retrieval of partial documents, *Information Processing and Management*, Vol. 31, No. 3, 1995, pp. 361-377.