EXTRACTING MAIN CONTENT-BLOCKS FROM BLOG POSTS

Saiful Akbar[1,2], Laura Slaughter[1] and Øystein Nytrø[1]
[1]*Department of Computer and Information Science*
Norwegian University of Science and Technology (NTNU), Trondheim, Norway

[2]*School of Electrical Engineering and Informatics, Institut Teknologi Bandung (ITB), Bandung, Indonesia*

Keywords: Blog post, Main block, Informative block, Content extractor.

Abstract: A blog post typically contains defined blocks containing different information such as the main content, a blogger profile, links to blog archives, comments, and even advertisements. Thus, identifying and extracting the main/content block of blog posts or web pages in general is important for information extraction purposes before further processing. This paper describes our approach for extracting main/content block from blog posts with disparate types of blog mark-up. Adapting the Content Structure Tree (CST)-based approach, our approach proposed a new consideration in calculating the importance of HTML content nodes and in definition of the attenuation quotient suffered by HTML item/block nodes. Performance using this approach is increased because posts published in the same domain tend to have similar page template, such that a general main content marker could be applied for them. The approach consists of two steps. In the first step, the approach employs the modified CST approach for detecting the primary and secondary markers for page cluster. In the next step, it uses HTMLFilter to extract the main block of a page, based on the detected markers. When HTMLFilter cannot find the main block, the modified CST is used as the second alternative. Some experiments showed that the approach can extract main block with an accuracy of more than 94%.

1 INTRODUCTION

The blogosphere is one means for patients/caregivers to write their daily journal about their own/patient's health problems and share the information with others. These blogs contain a huge amount of information, either medical- or non medical-related problems, which can be worth something for other patients, caregivers, or clinical researchers who are searching specific health issues. As an example, a person published his/her experience of living with her mother who has Lewy Body disease in LewyBodyJournal.org and the publication is represented in a 50-page ebook.

The blogosphere is one example of a collection of web pages that are published in a systematic standard format. They are automatically generated using a pattern, where the pattern is a set of HTML tags (Tseng & Kao 2006). The pattern divides the web page into several blocks, each with a specific purpose. A blog page may contain a blogger profile block, an archives links block, an advertisement block, a post/main block, a comment block and other blocks. Figure **1** shows a blog page with the main block, a comment block, an archieves link block, and a calendar block. In order to extract valuable information from the blog post, we need to distinguish a block, especially the main block, from the others.

Our research focuses on analyzing and extracting valuable information from blogs written by alzheimer patients or their caregivers. The paper discusses our initial step on the research, i.e. extracting the main content of alzheimer/dementia (AD) blog posts. We adopted and modified an approach based on Content Structure Tree (CST) proposed by 0. Although the extraction method described in this paper is tested only for alzheimer blog posts, it is reasonably applicable for general blog posts.

The rest of this paper is organized as follows. Section 2 discusses the classification of and some methods of extracting the main/content block, including the CST-based approach. Sections 3 presents our modification and extension of the CST

Akbar S., Slaughter L. and Nytrø Ø..
EXTRACTING MAIN CONTENT-BLOCKS FROM BLOG POSTS.
DOI: 10.5220/0003091804380443
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2010), pages 438-443
ISBN: 978-989-8425-28-7
Copyright © 2010 SCITEPRESS (Science and Technology Publications, Lda.)

approach: mCST and E-mCSTE-mCST. Section 4 presents experiments,performance comparisons between the methods, and evaluations. The conclusion is presented in Section 5.

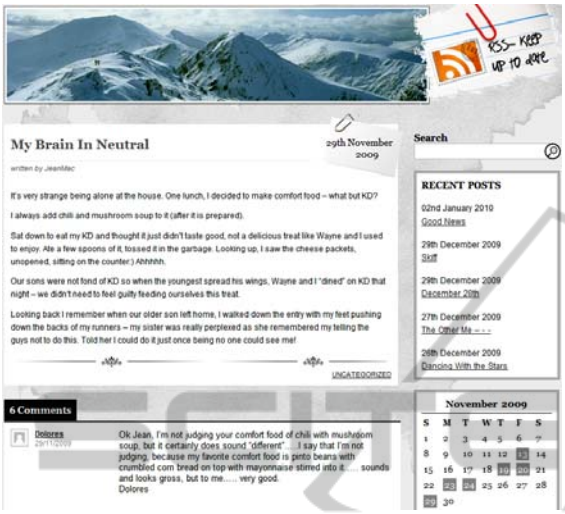

Figure 1: A blog post and its blocks.

2 RELATED WORK

In general, there are 3 approaches to extracting main block content from web pages: the static rule-based approach, the DOM-based approach, and the machine learning (ML) approach. Another approaches are basically built as a combination of some of the three approaches.

In general, rule-based approaches attempt to extract main content by establishing a rule associated with how the author of the blog or the blog builder software organizes and divides the blocks in a blog post. The approach is usually time-consuming since the rules are built manually. It also does not dynamically adapt new cases since the rule is designed to work for some specific blogs. In order for it to be dynamically adaptable, the rule should be built in a generalized form using such as pattern matching rule. An example of this approach is implemented by Attardi and Simi (2006). They extract the main block by identifying specific HTML elements that mark the main block, e.g. `<div class="post">` for blogs published through `WordPress`. In order to filter out non-relevant blocks, such as the navigation block, they identify elements that mark the non-content block. The main block is extracted by filtering in the part marked by a main content marker while out the parts marked by non-relevant block markers.

The second approach, DOM-based approach, focused on the observation of the tree structure of HTML document (i.e. the Document Object Model/DOM). In general, in order to determine the main/content block, the approach builds the tree structure of a page and evaluates the importance of each block,which is represented as a node in the tree structure. Then, the block with the highest block-importance value is determined as the main/content block. Examples are approaches proposed by Tseng & Kao (2006) and Li & Yang (2009). Tseng & Kao (2006) proposed some features for measuring the importance of a block, i.e.: regularity of node group, the density of a block, and the distribution of items style of objects. The importance of a block is defined as the product of the three measures. Different from Tseng & Kao (2006), Li & Yang (2009) employ the attenuation quotient and the importance of HTML item and content item. The attenuation quotient is used to decrease the node importance when a node is closer to the root. In other words, the deeper position of a node in the three, the bigger the attenuation is. The importance value of content item is defined as the number or the length of text, images, and links. Then, the importance of an HTML item node is defined by calculating the sum of attenuated-importance of its descendants and the content items. A more detailed description of the approach will be presented in Section 0.

A machine learning approach has been proposed by Lin & Ho (2002). They observe most of dotcom web sites use `<table>` as a layout template, and pages in the same domain contain redundant identical blocks (which is non-content block) and distinguishable blocks (which is considered as informative content block). By observing the tag `<table>`, a page is partitioned into several blocks. Thus, features of each block are extracted. In a page cluster (cluster with pages from the same domain), the entropy values of block features are calculated. A block with entropy less than a defined threshold is considered as an informative block since it is distinguishable from others (Lin & Ho, 2002). A similar approach has been proposed by Debnath et. al. (2005). The approach employs two algorithms: FeatureExtractor (FE) and ContentExtractor (CE). FE extracts text, link, and image features from block candidates and employs K-mean clustering to identify k blocks in an HTML page. CE observes some HTML pages from the same domain and calculates the similarity between each block in a page and other blocks in another page. Based on the fact that many pages in the same domain have

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

redundant blocks, except the main block, if two blocks are very similar to each other, they are considered as redundant blocks, and hence are not the main/informative blocks.

The work presented in this paper adopts and modifies the CST approach proposed by Li & Yang (2009). Different from the approach (Li & Yang, 2009), our approach employs the following strategy:

1. Employing HTML cleaner before processing pages.
2. Using only text for calculating block significance
3. Utilizing both the depth and number of a node's children as the attenuation quotient factor.
4. Employing two steps of main block detection. In the first step, modified CST is used for detecting primary and secondary main block markers of post from the same domain, while in the second step the main block is detected by filtering the post using the markers and by using the modified CST as the second alternative.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

3 EXTENDED CST APPROACH

The CST approach (Li & Yang, 2009) consists of two steps: (1) building the CST of an HTML page, and (2) measuring the importance of nodes which represents nested blocks, and determining the block with the highest importance value as the main block of the page. The CST contains two types of nodes: HTML item node and content node types. The former is generated from `<body>`, `<div>`, or `<table>` tags and represents blocks, while the later represents the actual content of the page and is generated from text, `<image>`, or `<link>` tags.

Starting from the deepest node, the importance of an HTML item node *N* is recursively defined as (Li & Yang, 2009):

$$\mu_N = \gamma_N . \mu_{C_N} + \sigma_{C_N} \quad (1)$$

where γ_N is the attenuation quotient of the node, μ_{C_N} is the importance of the children of the node N, and σ_{C_N} is the importance of HTML item nodes which are the children of node *N*, σ_{C_N} is the importance of content nodes which are the children of the node N and is calculated by counting the weighted sum of the size of text, the number of images and the number of links. The experiment reported by Li & Yang (2009) has shown that the optimal weight is 1.0, 0.75, and 40 for text, links, and images respectively. For a more detailed explanation the reader should refer to the original article (Li & Yang, 2009).

Based on the CST approach, we proposed two improvements: the modified CST approach (mCST) and the extended mCST (E-mCST). Our modification of the CST approach (the mCST approach) includes the following:

(1) We observe that in a blog post, a block is represented by not only `<body>`, `<table>`, `<div>`, but also other tags such as `<id>` and `<span>`. For simplicity, in our implementation we consider any tags other than text, image, and link tags as HTML item tags.

(2) Note that in our research case we need to extract the textual information from blog posts. Moreover, we observe that the main block of blog post usually contains very limited links and images. Hence, we propose to use text as the only factor for calculating the importance of content node.

(3) In the original CST approach (Li & Yang, 2009), the attenuation quotient is defined as:

$$\gamma_N = \begin{cases} 1, & D_N \le 10 \\ \log(D_N), & D_N > 10 \end{cases} \quad (2)$$

where D_N is the depth of the node *N*. The definition does not distinguish between a wide substree and a narrow subtree. Figure 2 ilustrates the situation. For clarity, circles represent item nodes and round-edge rectangle represent content nodes.

As shown in Figure 2, a node with narrow subtree contains more dense content than the one with wider subtree, therefore the the former should suffer from attenuation less than the later. Hence, considering thatbthe attenuation quotient should be influenced by not only the depth of the node but also the wide of the subtree of the node, we propose a new definition of attenuation quotient as:

$$\gamma_N = \frac{1.0}{Log(D_N + 10)} \cdot \frac{1.0}{Log(10 C_N)} \quad (3)$$

where D_N is the depth of the node *N*, and C_N is the number of the children of the node *N*.

In the E-mCST approach, we would also like to take the benefit of the fact that posts of the same domain tend to have similar block layout, which is represented by similar HTML item nodes. Thus, for posts from the same domain of address, we identify the HTML tags and their attributes that are

considered as the content block markers for the group of posts. Figure 3 shows the steps of the E-mCST approach).

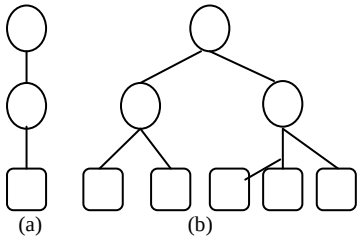

Figure 2: (a) Narrow, and (b) wide trees.

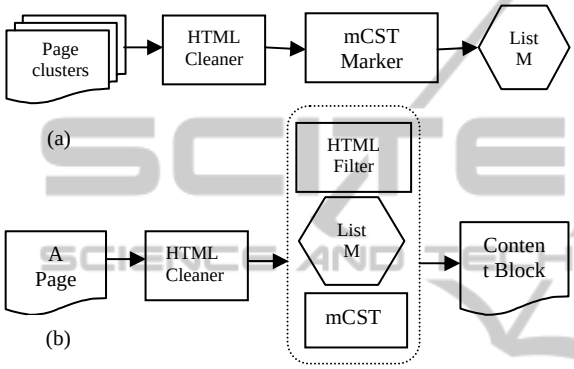

Figure 3: The Extended mCST (E-mCST) approach.

As depicted in Figure 3, we employ two steps: (a) marker detection and (b) content block extraction. In both steps, we employ HTMLCleaner[1] for cleaning up the page from unnecessary components such as HTML comments and scripting. It also used to obtain a well-formed HTML page before further processing.

In the first step (Figure 3.a), the input of the subsystem is page cluster, i.e. posts in the same address domain, and the output is marker list M. This subsystem used mCST (mCST) as a *marker detector* to identify tags and attributes considered as content block markers. From a page clusters we extract maximum 2 alternative markers (primary and secondary markers) that are most commonly identified by mCST.

For example, suppose that we obtain markers identified from a page cluster as depicted in Table 1. In the example, markers that are determined as the primary and the secondary markers are `div|class|entrybody` (found 3 times) and `div|class|snap_preview` (found twice), respectively. Thus, both markers are considered as the content block markers for the corresponding cluster. The detail of the algorithm for detecting markers is shown in Figure 4.

Table 1: An example of markers identified from a page cluster.

Page#	Identified Markers
P1	`div|class|snap_preview`
P2	`div|class|entrybody`
P3	`div|class|entrybody`
P4	`div|class|snap_preview`
P5	`div|class|outer`
P6	`div|class|entrybody`
P7	`Body`

```
Algorithm 1: mcST_MarkerDetector
Input: Set S of HTML pages
       of the same domain
Output : Primary and Secondary content
       block markers, pCM, sCM
Begin
   for each page H in S
      B ← mcSTExtractMainBlock(H)
      m ← GetTheRootMarker(B)
      if (m is unique in H)
         PutInList(m,M)
   pCM ← GetMarkerWith1stMaxOccurence(M)
   sCM ← GetMarkerWith2ndMaxOccurence(M)
   → {pCM, sCM}
End
```

Figure 4: The algorithm for detecting markers.

In the second step (Figure 3.b), we employ HTML Filter[2] based on list M for extracting the content block of a page. If the domain of the input page is found in the M list, HTMLFilter will use the first marker to filter the page. If the first marker is not found in the page, then the second marker is employed. If none of the marker is found in the page or the domain of a page is not found in the list, then mCST is employed.

4 EXPERIMENTS AND EVALUATIONS

In order to have the data set for the experiments, using Google search engine we search blogs that most likely contain text about alzheimer/dementia (AD) disease. We manually observe the blogs in order to decide whether the blogs contain significant number of posts. We also observe the links in the blogs to find other potential blogs. Using HTTrack

[1] http://htmlcleaner.sourceforge.net/

[2] HTML Filter is built using HTMLParser http://htmlparser.sourceforge.net/

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Website Copier[3] we download the blogs and get the post pages of the blogs. In total, we have 8270 posts from 48 blogs. We separate the blogs into two datasets: (1) well know blogs, i.e. blogs published in `wordpress.com` and `blogspot.com`, and (2) others. The first dataset contains 4713 posts of 38 blogs, while the second contains 3557 post of 10 blogs

In order to build the ground truth (the golden standard), we manually inspect some posts from each blogs and identify marker tags and build a rule for extracting the correct block content. There are two types of marker tag: IN marker and OUT marker. IN marker marks a block that should be taken as the content block, while the OUT marker marks a block nested in the block marked by IN marker that should be excluded from the content block. Together with HTMLFilter, the rule is used to filter in the content block and filter out some irrelevant nested blocks.

Figure 5 shows an example of rule for filtering in/out page content from posts with address domain `wordpress.com`.

```
(
      addr    =.*\.wordpress\.com.*

      in   = div|id|content-main
      out  = p|class|postmetadata
      out  = div|class|post-content
      out  = div|id|respond

      in   = div|id|content
      out  = div|class|navigation
      ……..
      out  = div|id|respond
)
```

Figure 5: A rule for filtering in/out page content.

In order to measure the performance of the approach, we compare text in the content block automatically extracted by employing the approach with the ground truth. We use cosine similarity for comparing the text. In order to measure the global performance of the approach, we employ two measurements: (1) ACS: the average of cosine similarity and (2) TCS: threshold based cosine similarity. In the first measurement, we calculate cosine similarity between a content block and its corresponding content block in the ground truth. Then, the average of cosine similarity is calculated. In the second measurement, if the similarity between an automatically extracted main block and its corresponding block in the ground truth is greater than a given threshold, then the two content blocks are accepted as identical. Then, the TCS total performance is defined as:

$$\rho = \frac{I}{N} \tag{4}$$

where I is the number of instances that are accepted as identical with the corresponding instances in the ground truth and N is the total number of instances in the dataset. In this experiment we used the threshold value of 0.9.

In order to observe the performance of each improvement step, we implemented and experimented with four methods as depicted in Table 2. First, we implemented the CST approach as proposed by Li & Yang (2009), used the same parameter, but used different definition of the HTML item node. In our implementation (see CST* in Table 2), an HTML item node is defined as a node representing any HTML tags other than image tags, link tags, and text. We consider this method as the baseline for the experiments. Second, we employ HTMLCleaner before employing the CST*. The HTMLCleaner not only removes unnecessary parts of an HTML page such as comments and scripts, but also more importantly transforms the page into a valid and well-formatted HTML page. The third implementation (mCST) differs from the second in two ways: (1) using text as the only factor for calculating the importance of content nodes, and (2) using the new attenuation quotient as described in Section 0. Finally, we implemented the extended CST approach (E-mCST) which is illustrated in Figure 3.

The performance comparison among the 4 different methods is shown in Table 2. As can be seen in the table, the use of HTMLClean outperforms the baseline, and the mCST approach outperforms the CST*+HTMLCleaner approach. It is also important to highlight that the E-mCST approach delivers the best performance among the others. We observe that by generating and employing filtering rule for specific domains, the E-mCST approach is still able to detect main blocks containing only little text, while the other approaches is not. Therefore we propose to use the E-mCST approach for extracting the informative block of blogposts.

It is interesting to see whether or not different domains yield significantly different performance. In order for that, we measure the performance of E-mCST for each cluster/domain as shown in Table 3.

[3] http://www.httrack.com/

Table 3 shows that in general E-mCST delivers very high accuracy (i.e. ACS > 95%, and TCS>93%), except for the three domains: `zarcrom.com`, `judyscaregiver.com`, and `yellowwallpaper.net`. The lower performance suffered by the three domains is due to the approach cannot detect/remove irrelevant blocks nested in the main block. This will be one of the issues for the next improvement.

Table 2: Performance comparison among the different methods.

Dataset	**Dataset #1 (4713 posts)**		**Dataset #2 (3557 posts)**	
Measure	**ACS**	**TCS**	**ACS**	**TCS**
CST*	0.878	0.445	0.900	0.642
CST*+HTMLCleaner	0.908	0.631	0.902	0.656
mCST	0.940	0.797	0.957	0.797
E-mCST	0.982	0.944	0.980	0.956

5 CONCLUSIONS

We proposed an extended CST approach (E-mCST) to identify the main/content block of blog posts. We modified the calculation of node importance proposed by Li & Yang (2009), by ignoring links and images and using text as the only factor for calculating content node importance. We also considered both number of children of a node and the depth of the node as factors for attenuation quotient. The approach consists of two steps. In the first step, the modified CST (mCST) is used to identify the primary and secondary content block markers of posts of the same domain, while in the second step the primary and secondary content block marker is used for filtering and extracting the content block, together with mCST as the alternative. The experiments have shown that the extended CST-based approach (E-mCST) outperforms the CST-based content block detector.

Table 3: Performance of E-mCST for each domain.

Domain	**Posts#**	**ACS**	**TCS**
wordpress.com	420	0.971	0.938
blogspot.com	4293	0.983	0.944
zarcrom.com	*54*	*0.944*	*0.741*
lewybodyjournal.org	24	0.999	1.000
judyscaregiver.com	*28*	*0.953*	*0.714*
mydementedmom.com	54	0.988	1.000
themomandmejournalsdotnet.net	1970	0.977	0.944
yellowwallpaper.net	*111*	*0.927*	*0.784*
blog.fadingfrommemory.info	329	0.998	1.000
annerobertson.com	165	0.998	1.000
thetripover.com	531	0.983	1.000
amountaintoohigh.com	291	0.990	1.000

REFERENCES

Tseng, Y. F., Kao, H. Y., 2006. *The Mining and Extraction of Primary Informative Blocks and Data Objects from Systematic Web Pages*. Proceeding of the 2006 IEEE/WIC/ACM International Conference, pp. 370-373.

Attardi, G., Simi, M., 2006. *Blog Mining Through Opinionated Words.* Proceeding of Text Retrieval Conference.

Li, Y., Yang, J., 2009. *A Novel Method to Extract Informative Blocks from Web Pages*. International Joint Conference on Artificial Intelligence, pp.536-539.

Lin, S. H., Ho, J. M., 2002. *Discovering Informative Content Blocks from Web Documents*. SIGKDD, pp 588 – 593.

Debnath, S., Mitra P., Giles, C. L., , 2005. *Automatic Extraction of Informative Blocks from Webpages.* Proceedings of the 2005 ACM symposium on Applied computing, pp. 1722 – 1726.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS