Journal of King Saud University – Computer and Information Sciences (2014) **26**, 19–27

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

CaSePer: An efficient model for personalized web page change detection based on segmentation

K.S. Kuppusamy *, **G. Aghila**

Department of Computer Science, School of Engineering and Technology, Pondicherry University, Pondicherry, India

Received 6 June 2012; revised 12 December 2012; accepted 27 February 2013
Available online 7 March 2013

KEYWORDS

Web page change detection;
Web page segmentation

Abstract Users who visit a web page repeatedly at frequent intervals are more interested in knowing the recent changes that have occurred on the page than the entire contents of the web page. Because of the increased dynamism of web pages, it would be difficult for the user to identify the changes manually. This paper proposes an enhanced model for detecting changes in the pages, which is called CaSePer (**Ch**ange detection based on **Se**gmentation with **Per**sonalization). The change detection is micro-managed by introducing web page segmentation. The web page change detection process is made efficient by having it perform a dual-step process. The proposed method reduces the complexity of the change-detection by focusing only on the segments in which the changes have occurred. The user-specific personalized change detection is also incorporated in the proposed model. The model is validated with the help of a prototype implementation. The experiments conducted on the prototype implementation confirm a 77.8% improvement and a 97.45% accuracy rate.

1. Introduction

The World Wide Web has evolved as a colossal content source that holds information on almost all the fields that are known to humanity. Human knowledge is evolving every day, and similarly the World Wide Web is developing. The contents of the World Wide Web exhibit an increasing level of dynamism. The World Wide Web has evolved from a simple and static content source into a richly dynamic information delivery channel. Hence, tracking the changes in web pages has become an interesting research problem; solving this problem would facilitate users to efficiently utilize the information that is provided by web pages.

Changes in web pages can either be content changes or structural changes (Flesca and Masciari, 2003). The structural changes are primarily changes that occur at the template level. Changes in the placement of elements are also covered under structural changes. The content changes include modification, insertion or deletion of hypertext elements. Users navigate to web pages that they are interested in. A user interested in a specific topic might visit web pages that are related to that topic at regular intervals over time. Such users might be interested in knowing the recent changes that have occurred in that web page, rather than seeing the entire web page. Change detection in web pages is a complex process. Because of the large number

* Corresponding author. Tel.: +91 4132655048.
E-mail addresses: kskuppu@gmail.com (K.S. Kuppusamy), aghilaa@yahoo.com (G. Aghila).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

1319-1578
http://dx.doi.org/10.1016/j.jksuci.2013.02.001

of nodes in an HTML tree, change detection requires many comparisons, which lead to an NP-hard problem (Chawathe and Garcia-Molina, 1997; Chawathe et al., 1996).

This paper aims to present an efficient change detection technique by splitting the problem into two major steps. The objectives of this research study are as follows: to propose an efficient model called CaSePer for web page change detection using web page segmentation; and to improve the change detection process by emphasizing user-specific informational needs.

Web page change detection can be utilized in various applications such as web page archiving, temporal querying, indexing, crawling and temporal visualization. The proposed CaSePer model is designed as a general purpose model to allow it to be adopted for domain-specific purposes. The remainder of this paper is organized as follows: In Section 2, some of the related studies that were accomplished in this domain are explored. Section 3 addresses the proposed CaSePer model's mathematical representation. Section 4 is about a prototype implementation and experiments. Section 5 focuses on the summary and conclusions derived from this research work.

2. Related studies

This section highlights the related studies that have been conducted in the same domain. The proposed model incorporates the following two major fields of study: web page change detection and web page segmentation.

2.1. The web page change detection

Change detection in the World Wide Web is the process of identifying both structural and content-based changes in the web pages. Change detection is an active research area, and there are many research studies in this domain. The studies on web page change detection analyze web pages in the temporal dimension. The temporal dynamism that is exhibited in the web contents is the focus of all the studies that have been conducted in the web page change detection domain.

AT&T Internet Difference Engine (AIDE) is a change detection system that employs a token-based approach (Douglis et al., 1998). AIDE works on HTMLDiff (Douglis and Ball, 1996) which is based on the Longest Common Subsequence (LCS) algorithm (Daniel, 1997). AIDE displays the changes between the current version and the immediate last version. The AIDE system compares different versions of a complete web page to detect the changes, whereas the proposed CaSePer model reduces the problem space by narrowing down the comparison region without compromising the quality of the change detection process.

The change detection approach that considers only the latest two versions of the page was given by WebCQ (Liu et al., 2000). At the same time, the WebCQ incorporates user-specific notifications. The proposed CaSePer model incorporates the personalization process and multi-hop signature matrix to consider various versions of the page. A dataflow-based approach to "change detection" was proposed in WebVigiL (Jacob et al., 2004). One of the important parameters in the change detection system is the revisit time interval. The revisit interval can either be given explicitly or can be learned by the system automatically. The WebVigiL system facilitates both of these approaches. The proposed CaSePer model incorporates a category-based revisit interval policy.

An improved change detection system that restricts the number of comparisons has also been explored (Artail and Fawaz, 2008). This study states that change detection can be performed by comparing only the similar tag types and by utilizing hashing. The proposed CaSePer model employs web page segmentation followed by a hashing technique to reduce the number of comparisons significantly. The "change detection" is attempted with the help of "edit-script", which refers to steps that are involved in converting one version of the document tree to another. The XyDiff (Cobena et al., 2002) algorithm provides a solution for the change detection problem using edit scripts. However, the sequence generated by XyDiff need not be optimal always. The X-Diff (Wang et al., 2003) ensures the optimal differences but it cannot handle the move operations. Typically the change detection approaches create a change representation file called a delta file. The delta file visualization using style sheets was proposed by La Fontaine (2001).

The proposed CaSePer model utilizes a variation of X-Diff in combination with a node sequence-based algorithm to handle both the content and structural changes, which is explored further in Section 3.1.

2.2. Web page segmentation

Web page segmentation is an active research topic in the information retrieval domain. Web page segmentation is the process of identifying segments that are sub-parts of a page. These smaller sections of the webpage are called segments. Segments are defined as follows (Chakrabarti et al., 2008):

> "Segment is a fragment of HTML, which when rendered, produces a visually continuous and cohesive region on the browse window and has a unified theme in its content and purpose"

The process of identifying these segments is called segmentation. The four basic types of web page segmentation methods are fixed-length page segmentation, DOM-based page segmentation, vision-based page segmentation and combined/hybrid page segmentation. A comparative study among various types of segmentation is analyzed in detail in the literature (Yesilada, 2011). Fixed length page segmentation is simple and less complex in terms of implementation, but the major problem with this approach is that it does not consider any visual rendering semantics of the page while segmenting it. Because the CaSePer model addresses web content and structural changes, this method is not adopted.

In DOM-based page segmentation, the HTML tag tree's Document Object Model would be used while segmenting (Buyukkokten et al., 2001). The CaSePer model harnesses the DOM- based segmentation as a component in its hybrid segmentation approach. The reason for not using the DOM tree approach alone is that, at times, improperly constructed HTML pages would fail completely in this approach.

The way that the VIPS algorithm views the web page is in parallel with the way that a human views it. VIPS (Cai et al., 2003) is a popular segmentation algorithm that segments a page based on various visual features. This approach harnesses the visual cues of the page and performs segmentation based on the visual representation of the page. Although the VIPS

approach incorporates visual rendering and provides improved results, the segmentation in the CaSePer model is targeted toward change detection and, hence, it does not require a visual segmentation technique. The primary goal of performing, the segmentation in the proposed model is to reduce the problem space and hence, a non-visual simpler segmentation approach is proposed.

Apart from the above-mentioned segmentation methods, a few novel approaches have evolved during the last few years. The term segmentation is widely used in the image processing domain as well. The segmentation of images differs from web page segmentation. An image processing-based segmentation approach utilizes (Cao et al., 2010) various edge detection techniques to find the segment boundaries of a web page. The downside of this approach is that the entire page is considered as an image and no semantics of the page is considered during the segmentation process. Web image context extraction techniques have also been explored in recent studies (Alcic and Conrad, 2011, 2010; Fauzi et al., 2009).

The segmentation process based on the text density of the contents has also been explored in a recent study (Kohlschütter and Nejdl, 2008). In this approach, the segmentation is performed based on the amount of text that is present in a two-dimensional area. This geometrical approach toward segmentation utilizes the density of the text as a deciding factor. Based on this density, the page is split into various segments. The CaSePer model incorporates densitometry as another component in its hybrid approach in addition to the DOM tree approach. The capability of marking the segment boundaries based on the change in the density slope is the rationale behind incorporating this approach in the proposed model.

Web page segmentation is also employed to detect changes in the web pages. The Vi-Diff (Pehlivan et al., 2010) has proposed an approach for change detection in web pages by incorporating web page segmentation. The Vi-Diff model extends the VIPS (Cai et al., 2003) algorithm for segmentation. In another study (Kukulenz et al., 2008), grammar-based decomposition of pages into segments is adopted to detect the changes.

The proposed CaSePer model utilizes the hybrid segmentation approach, amalgamating the DOM page tree and densitometry techniques. The rationale behind the hybrid approach incorporating DOM tree and densitometry is that this approach addresses the segmentation problem by considering the placement of elements in the tree and the density with which the contents are placed. The model harnesses the density slope difference value to fix the segment boundary. The CaSePer model also incorporates a node boundary and cascaded node sequences in the segmentation process, for successive versions of the page, which is explored further in Section 3.1.

3. The CaSePer model

This section explores the proposed CaSePer model of web page change detection. The proposed model handles the complex change detection process by splitting it into two major steps: segmentation of the web page into smaller components and computing the hash value on these components. The segments with non-identical hash values with their temporal counterparts are promoted for the next step, which performs the actual comparison process. After locating the segments, which have non-identical hash values compared with their temporal counterparts, the precise changes are identified by adapting the edit script and node sequence-based tree comparison techniques.

The block diagram of the proposed CaSePer model is shown in Fig. 1. The model includes the following components: the Page Evolution Track holds snapshots of the pages that were taken at different time intervals. The proposed model assumes an index structure that builds the snapshots of the page at different time intervals and stores it in the Page Evolution Track. The Segmentor component performs the web page segmentation task using a hybrid segmentation approach and incorporating the Document Object Model and densitometric approaches. Hash Builder is responsible for calculating the hash value of individual segments. The Hash Comparator's role is to identify the set of segments that have changes by comparing their hash values. The Target Segment Pool holds the segments that are identified with change. The Segment Comparator performs the actual comparison, to detect content and structural changes. The User Profiler holds the profile bag, which contain the user profile keywords. The Term Expander module performs the task of expanding the user profile keywords by fetching additional terms from ''wordnet''. The Term Highlighter is responsible for highlighting the general changes with style and profile-specific keywords, with another unique style for easy identification by the user. The CaSePer model detects both the general and user specific changes in the web page by using the above specified components.

The Segmentation method employed in the proposed model belongs to the hybrid segmentation type. A combination of Document Object Model and Densitometry based approaches is harnessed in the proposed model for segmenting the page into smaller blocks. The segmentation process is performed on the server side, to achieve a higher speed of execution, which is a vital factor in the web scenario.

The CaSePer model proposes a bi-fold segmentation technique. The initial version of the web page is segmented with the help of a server side, bottom–up segmentation technique that functions as follows: The Document Object Model tree of the page is constructed, and the tree is navigated in a bottom–up manner, in the decreasing order of the tree depth. The nodes at each level of the tree are identified either as ''block-level nodes'' or ''non-block level'' nodes. The block-level nodes are the nodes that have the capability of holding other child nodes in them. The non-block level nodes are atomic elements that cannot hold child elements in them. For the block level nodes the tree size is computed, and if the size exceeds a threshold value that can be set according to the nature of the web page collections, then the densitometry technique is applied. If the size is less than the threshold value then the tree is merged with its predecessors. For each subtree that is identified with a size of more than the threshold limit, the contents of the subtree are wrapped in fixed length lines, and the text density is computed for each line. The slopes among the text densities of the successive lines are checked with the slope-threshold, and if a slope exceeds the threshold, then each of the candidate segments are considered to be individual segments; otherwise, they are merged into a single segment and the procedure is repeated. In the case of non-block level nodes, if their size exceeds the threshold limit, then they are considered to be separate segments; otherwise, the non-block nodes are merged with the nearby block level or non-block level nodes.

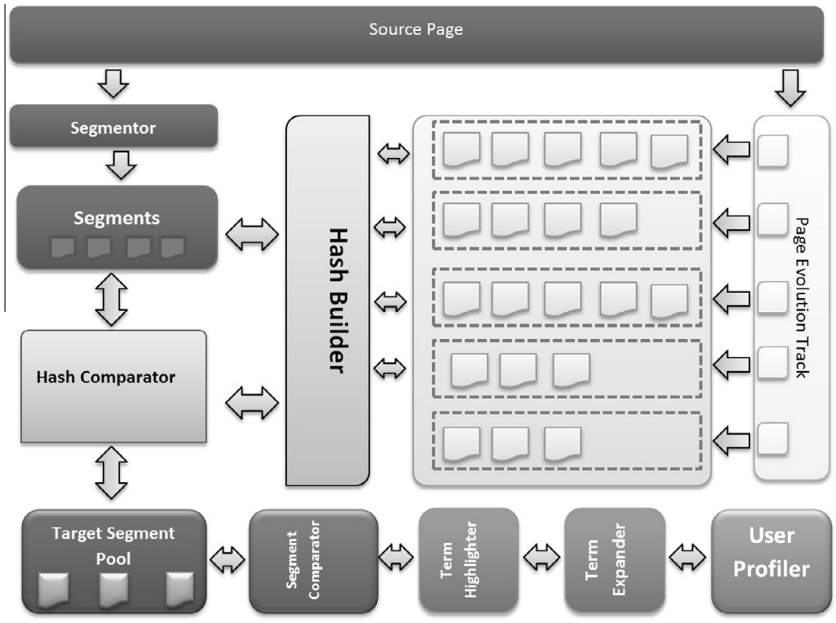

Figure 1 Block diagram of the CaSePer model.

For the successive versions of the same web page, segmentation is performed by using the node boundary-based approach. This bi-fold segmentation technique is proposed by CaSePer so that the number of segments in all the versions of the web page would be kept the same, which makes the change detection efficient. In this technique, the boundaries of segments are identified with the help of DOM tree nodes, which formed the boundaries in the previous version. Because the boundary nodes themselves can become removed or relocated, a cascaded node sequence from the segments of earlier versions are used to mark the segment boundaries. The sequence of nodes in the earlier versions is utilized to mark the segment boundary. As the sequence can change across versions because of this structural modification, this approach is adopted in combination with the boundary node. This method is followed to allow the number of segments in all the versions of the page to remain the same.

The above specified bi-fold segmentation method builds a segment pool from the web page. Each of the segments computes a hash value. In the proposed implementation, the hash function chosen is the MD5 (Rivest, 2008). The reason for selection of MD5 is that it will never produce the same hash value for two inputs, even with the slightest variation in the data.

The segments with non-identical hash-values along the temporal dimension are selected for a further change-detection process. If the segments have identical hash values then the corresponding segment is not modified in the versions that are under comparison.

Because the segments that are identified with the hybrid segmentation approach are considerably smaller in size, the changes that occur in the segment can be located precisely, and the comparison is made much simpler. To compare these treelets of the segments and to detect the changes, a variation in the X-Diff (Wang et al., 2003) algorithm is used to detect the content changes, and node path sequences are used to detect the node repositioning, which becomes reflected as structural changes in the rendering. The content change detection technique is based on the edit script, which is a sequence of basic edit operations that converts one tree into another (Chawathe et al., 1996).

The edit distance refers to the number of inserts, deletes and update operations that are required to convert one tree into another. The DOM sub-trees that correspond to each segment are called treelets. Because the proposed CaSePer model operates at a fine-grained segment level, the size of individual treelets is considerably smaller. The nodes of the treelets in the versions of web pages are analyzed with the edit distance for content changes. The X-Diff algorithm can detect changes in an unordered manner, which does not consider the ordering among the sibling nodes. However, with the web page, if the ordering of siblings becomes altered, the rendering of the page would also be modified. To identify these structural changes, the node paths are compared to find any relocation of nodes.

The proposed method incorporates the following changes in the X-Diff approach: Instead of selecting the nodes for matching, on the basis of their signature, the proposed model compares the nodes based on their ID attribute and type. This comparison is feasible because of the smaller size of the treelets. To locate the structural changes, the node path sequence comparison is performed, whereas the X-Diff approach can detect only changes in an ancestral relationship.

The node path comprises a sequence of nodes from the ancestral starting point in the segment boundary to the current node. The individual nodes in the path are separated by a "|" symbol to construct the complete path. The path of a typical node looks like "tag1- id | tag2- id |...|tagn- id". If the node in the DOM tree does not have an "id" attribute, then it is left blank in the node sequence. Then, the paths of nodes in different versions are compared by considering the current version as the source and the temporal predecessor as the destination. The nodes with "id" attributes are compared for their index values in both of the node paths. The index value is assigned in a sequential manner, starting from the left side of the node

sequence and incrementing a value of one for each "|" symbol. If the index values are equal, then there is no change in the structure; otherwise, a change is detected. When a change is detected, then an offset value is added for the successive index values to nullify the effect of the current change. If this offset is not incorporated, then, after the first change, all the nodes would have different index values. For nodes that are missing the "id" attribute, the node's content hash value is generated; these nodes are then compared with counterparts that have the same hash value. If there is no match in the hash value, then these nodes are filtered out from both of the sequences and are compared using their node type. Because the treelets are smaller in size, the comparison of the nodes of the same type becomes less complex.

The above specified method of splitting the web page into smaller segments, computing their hash values, and identifying the segments that are changed greatly reduces the complexity of individually comparing each HTML node to every other HTML node. The incorporation of this segmentation step reduces the change detection problem from the NP hard status to a manageable complexity.

Another important feature of the proposed change detection model is the ability to highlight the user profile-specific term in the changes. This feature makes the model tailor-made to the specific interests of the user. This specificity is achieved by incorporating a user profile bag with keywords that were identified by the user. Apart from directly using these keywords, the extended profile terms are gathered with the help of lexical databases such as Wordnet (Fellbaum, 2010; Miller, 1995). The user profiles are gathered using various social information of the user, as in an earlier study, to customize the search results (Hassanpour and Zahmatkesh, 2012). User profiles are built in the proposed model, by adopting the FOAF (Friend Of A Friend) ontology based technique, which is used to hold preferences of the user in the form of XML representations.

The proposed CaSePer model adopts a FOAF (Fried of a Friend) based profile representation technique. The profile keywords are built using a multimodal approach (Kuppusamy and Aghila, 2013). The user supplies a FOAF file, which contains the user's interest terms, interest pages, blog and work-place-home-page. The personalized changes identified in the web pages are highlighted using a distinct style to allow the user to easily recognize the personalized changes from generic changes.

The mathematical representation of the above specified procedure is illustrated in the following section.

3.1. Mathematical model

The proposed CaSePer model for web page change detection initially splits the current web page into various segments. The model involves two different procedures for segmenting the web pages, as follows: A hybrid segmentation approach amalgamating DOM and densitometry is used to segment the initial version of the web page. The successive versions of the same web page are segmented using node boundary and cascaded node sequence techniques, proposed by the CaSePer model. The hybrid segmentation technique proposed in the CaSePer model identifies the block level and non-block level nodes. Then, in the block level, the text density is applied to perform segmentation.

Initially, the web page is parsed to locate block and non-block level elements, as indicated in Eq. (1). The identification of block and non-block level elements starts at the node with the largest depth value. This process is continued in the decreasing order of depth.

$$\Omega = \{\eta, \xi\} \tag{1}$$

In Eq. (1), η represents the block level nodes and ξ represents the non-block level nodes. The block-level nodes must be analyzed for their sub-tree structure and, if it is more than the specified threshold value, the block level element is considered to be an individual segment; otherwise, it must be merged. The block level elements and non-block level specifications proposed by the CaSePer model are in alignment with the W3C HTML 4.0 specifications.

$$\forall_{i=1;j=1}^{n,m}\omega_j = \left\{ \begin{array}{cc} \eta_i & \text{if} \quad \Re(\eta_i) \geqslant \delta;\ i++,\ j++ \\ \omega_{j-1} \cup \eta_i & \text{otherwise;}\ j++ \end{array} \right\} \tag{2}$$

In Eq. (2), $\Re(\eta_i)$ indicates the function for calculating the size of the sub-tree for the node η_i. If $\Re(\eta_i)$ is greater than the threshold limit δ, then η_i is assigned to segment ω_j Otherwise it is merged with the predecessor ω_{j-1}. These larger segments are then sub-divided using the densitometer. The text density and the slope between the blocks are defined by the densitometric segmentation process (Kohlschütter and Nejdl, 2008). In the proposed model, the densitometer analyzes individual blocks separately instead of considering the entire web page. The sub-tree text of $\Re(\eta_i)$ is analyzed by the densitometer. The source of the sub-tree is wrapped in a fixed length, and each resulting line is considered as a candidate segment.

$$\forall_{j=1}^{\sigma}\varepsilon[\Re(\eta_i)]_j = \left\{ \frac{tc([\Re(\eta_i)]_j)}{\sigma([\Re(\eta_i)]_j)} \right\} \tag{3}$$

In Eq. (3), $\left\{\frac{tc([\Re(\eta_i)]_j)}{\sigma([\Re(\eta_i)]_j)}\right\}$ represents the ratio of the token count in the j^{th} line of the sub-tree $\Re(\eta_i)$, to the number of fixed length lines that it consumes. The slope between candidate-segments j and $j+1$ are indicated below:

$$\Delta\varepsilon\{[\Re(\eta_i)]_j, [\Re(\eta_i)]_{j+1}) = \frac{|\varepsilon\{[\Re(\eta_i)]_j\} - \varepsilon\{[\Re(\eta_i)]_{j+1}\}|}{\max(\varepsilon\{[\Re(\eta_i)]_j, [\Re(\eta_i)]_{j+1})} \tag{4}$$

If the calculated slope value is greater than a threshold value λ then $[\Re(\eta_i)]_j$ and $[\Re(\eta_i)]_{j+1}$ are considered to be separate blocks ω', ω''; otherwise, they are fused together (ω), as shown in Eq. (5).

$$\left\{ \begin{array}{cc} \omega = [\Re(\eta_i)]_j \cup [\Re(\eta_i)]_{j+1} & \text{if } \Delta\varepsilon\{[\Re(\eta_i)]_j, [\Re(\eta_i)]_{j+1}) < \lambda \\ \omega' = [\Re(\eta_i)]_j & \\ \omega'' = [\Re(\eta_i)]_{j+1} & \text{otherwise} \end{array} \right\} \tag{5}$$

With respect to the web page change detection problem, the segmentation is performed only to reduce the complexity of the problem by narrowing down the search space. Hence, the segmentation process incorporated in the CaSePer model is oriented toward minimizing the time that is taken for segmentation rather than considering the visual features.

At the end of the segmentation process, the segment pool is filled with the segments that were identified by using the above-specified procedure, as shown in Eq. (6).

$$\Omega = \{\omega_1, \omega_2, \omega_3, \ldots \omega_n\} \tag{6}$$

Then, for each segment ω_i, a hashing function is applied to calculate the hash value.

$$\Psi = \{\forall_{i=1\ldots n}\delta_i = md5(\omega_i)\} \tag{7}$$

The previous snapshots of the web page Ω, in the temporal dimension, is fetched from the Page Evolution Track (PET) Γ. The Page Evolution Track is an indexing structure that has a temporal capability, which holds the snapshots of the previous versions of the web page.

$$\Gamma = \{\forall_{i=1\ldots m} t_i(\Omega)\} \tag{8}$$

In Eq. (8), each $t_i(\Omega)$ represents the snapshot of the page Ω taken at time t_i. The length of the page evolution track is represented by m.

The segment matrix of the snapshots in the Page Evolution Track Γ is built as shown in Eq. (9).

$$\Gamma_s = \begin{Bmatrix} \omega 1, t1 & \omega 2, t1 & \omega n, t1 \\ \omega 1, t2 & \omega 2, t2 & \omega n, t2 \\ \omega 1, tm & \omega 1, tm & \omega n, tm \end{Bmatrix} \tag{9}$$

In Eq. (9), each $\omega_{i,\ tj}$ represents the ith segment in the snapshot that was taken at time t_j. Individual rows in Γ_s represent a complete web page. The columns in Γ_s represent the evolution track of the page.

Then, for each element in Γ_s, the MD5 hash value is calculated, as shown in Eq. (10).

$$\Psi(\Gamma_s) = \begin{Bmatrix} \delta 1, t1 & \delta 2, t1 & \delta n, t1 \\ \delta 1, t2 & \delta 2, t2 & \delta n, t2 \\ \delta 1, tm & \delta 1, tm & \delta n, tm \end{Bmatrix} \tag{10}$$

The matrix represented by $\Psi(\Gamma_s)$ is called the "Signature Matrix". Each $\delta_{i,\ tj}$ in $\Psi(\Gamma_s)$ is derived using Eq. (7). Each $\delta_{i,\ tj}$ represents the hash value computed for the i^{th} segment at time interval j. The first row of the signature matrix $\Psi(\Gamma_s)$ represents the current version of the web page.

The hash value specified in the signature matrix is compared row-wise. If the hash values in the first row are different from the hash values in the successive row, then a change is detected. The segments holding non-identical hash values are the segments that have their structure or content changed. The segments that hold the newly-introduced hash values are extracted, as shown in Eq. (11).

$$\Omega_c = \begin{Bmatrix} \omega_{i,t1} & \text{if } (\forall_{i=2..n, j=2..m}\delta_{i,t1} \cap \delta_{i,tj} = NULL) \\ 0 & \text{otherwise} \end{Bmatrix} \tag{11}$$

The non-zero elements of Ω_c are the segments in which the changes have occurred. This step of identifying the segments that have undergone changes makes the model robust by shrinking the problem space boundary from a larger area into a focused sub-section, which obviously reduces the complexity in terms of identifying the change. The non-zero elements of the set Ω_c can be represented as shown in Eq. (12).

$$\Omega_c = \{\bar{\omega}_1, \bar{\omega}_2, \ldots \bar{\omega}_q\} \tag{12}$$

Each $\bar{\omega}_i$ represents the segments that have changed. These $\bar{\omega}_i$ are compared with their temporal predecessors for accurately detecting the change using edit distances.

To introduce personalization, into the process of change detection, the user profile-based terms are highlighted. The user profile is recorded as a collection of keywords, as shown in Eq. (13). Furthermore, the model adds on keywords by extracting key terms from the supplied resource identifiers.

$$\Theta = \{\alpha_1, \alpha_2, \ldots \alpha_n\} \tag{13}$$

To highlight the user-specific keywords, Ω_c is compared with Θ, as shown in Eq. (14).

$$\Upsilon = \frac{\{\bar{\omega}_1, \bar{\omega}_2, \ldots \bar{\omega}_q\}}{\{\alpha_1, \alpha_2, \ldots \alpha_n\}} \tag{14}$$

To enrich the profile-specific highlighting feature, the user-specified keywords are expanded by referring to "wordnet" and collecting the synonyms, hyponyms and hypernyms. This step makes the proposed model more efficient by detecting those changes that do not directly involve the user-specified profile term but instead involve one of their linguistic variations.

$$\vec{\Theta} = \{\alpha_1 = \{\pi(\alpha_1)\}, \quad \alpha_2 = \{\pi(\alpha_2)\}, \ldots \alpha_n = \{\pi(\alpha_n)\}\} \tag{15}$$

In Eq. (15), $\pi(\alpha_i)$ represents the collection of synonyms, hyponyms and hypernyms that are derived from "wordnet". Then, these extended profile terms are compared with the segments that have undergone change, as shown in Eq. (16).

$$\vec{\Upsilon} = \frac{\{\bar{\omega}_1, \bar{\omega}_2, \ldots \bar{\omega}_q\}}{\vec{\Theta}} \tag{16}$$

The changes identified in $\vec{\Upsilon}$ would include the extended profile terms as well. The proposed CaSePer model identifies the general changes and the specific changes that are associated with the user's specific information requirement context. This facility of identifying the user-specific changes in the web page would make it convenient for the user to easily locate the changes that could be relevant to one's interests, in a much simpler and efficient manner. At the implementation level, these user-specific changes are rendered with a unique style definition that is different from the general changes, to catch the user's attention.

4. Experiments and result analysis

This section explores the experimentation and results that are associated with the proposed CaSePer model for web content change detection using segmentation and personalization. The prototype implementation is accomplished with the software stack that includes Ubuntu Linux, Apache, MySql and PHP. For client side scripting JavaScript is used. With respect to the hardware, a Core i5 processor system with 3 GHz of speed and 8 GB of RAM is used. The internet connection used in the experimental setup is a 128 Mbps leased line.

The experiments are conducted with a custom built crawler that starts with a set of seed URLs and a "revisit interval flag" (RIF). The seed URLs are the locations from which to start the crawl, and the revisit interval flag holds the time interval between the successive visits to the same page by the crawler. The revisit interval flag is assigned a value that is based on the type of page. In the experiments, three different categories of seed URLs were used. They are "academic", "personal" and "news". For the academic category, the RIF is set at 48 h; for the personal category, it is set at 24 h, and for news 1 h. The rationale behind such an approach is that the frequency of modification gradually reduces from the news web

Table 1 Experimental results of the proposed model.

Session ID	MSC	MPETL	MSM	MSMP	MSUD	SDP
1	23.12	4.3	3.1	1.2	0.43	98.14
2	18.45	5.2	4.1	2.3	0.78	95.772
3	17.45	5.3	3.2	2.6	0.23	98.682
4	20.32	4.3	4.4	3.1	1.1	94.587
5	18.76	3.5	1.2	0.5	0.45	97.601
6	21.08	5.6	3.2	2.6	0.35	98.34
7	20.45	4.2	4.5	2.5	0.21	98.973
8	20.32	3.5	4.3	3.1	0.33	98.376
9	19.34	4.1	3.5	1.2	0.21	98.914
10	24.12	5.1	3.7	1.1	0.56	97.678
11	17.65	5.2	7.8	3.8	0.87	95.071
12	16.78	5.3	6.5	4.1	0.23	98.629
13	17.65	4.4	4.2	2.1	0.4	97.734
14	14.89	4.5	5.3	3.1	0.33	97.784
15	13.23	4.7	4.1	1.3	0.55	95.843

pages to personal to academic. The snapshots of the pages are stored in the Page Evolution Track by utilizing the Apache lucene indexing mechanism. The crawl gathered 8,544 unique pages from different domains, with an average page evolution track length of 4.6. The experiments were conducted in various sessions. Each session used 10 different pages from the crawl. This approach is adopted so that the efficiency of the approach is computed as a means of values across the sessions. The data from the results of the experiments are tabulated in Table 1.

In Table 1, MSC stands for the Mean Segment Count, which indicates the mean of the number of segments in that session. MPETL stands for Mean Page Evolution Track Length, MSM represents the Mean Segments Modified, MSMP stands for Mean Segments Modified with Personalized keywords, MSUD represents the Mean Segments Undetected and SDP represents the Successful Detection Percentage.

In Fig. 2, the chart depicts the MSM and the MSMP. The MSM with respect to general changes across the session is observed to be 4.2. The mean of the overall segment count in the experiments is 18.9. By the incorporation of segmentation-based hash value comparisons the number of segments to be considered by the tree comparison was reduced by 77.8%.

The MSMP across the sessions is observed to be 2.3. The segments with user-specific modifications were estimated to be 54.77% of the overall changes; these changes were rendered with a unique style for the easy identification of the user. This 54.77% emphasizes the importance of incorporating personalization into the change detection process.

The Mean of the Successful Detection Percentage across the session is 97.45, which validates the robustness of the proposed change detection model.

The chart in Fig. 3 depicts the time that was taken for various components of the CaSePer model such as segmentation, content change detection, structural change detection, profile term highlighting and rendering for web pages of various sizes. The chart depicts the mean values for experiments which were conducted in 15 sessions. In the *X*-axis the mean page size is plotted, and *Y*-axis denotes the time in seconds. The segmentation method adopted in the CaSePer model is optimized for speed rather than for the semantics.

It can be observed from Fig. 3 that the mean of the segmentation time is 0.2 s which confirms the efficiency of the segmentation process that is associated with the CaSePer model. With respect to the change detection, the time to detect changes

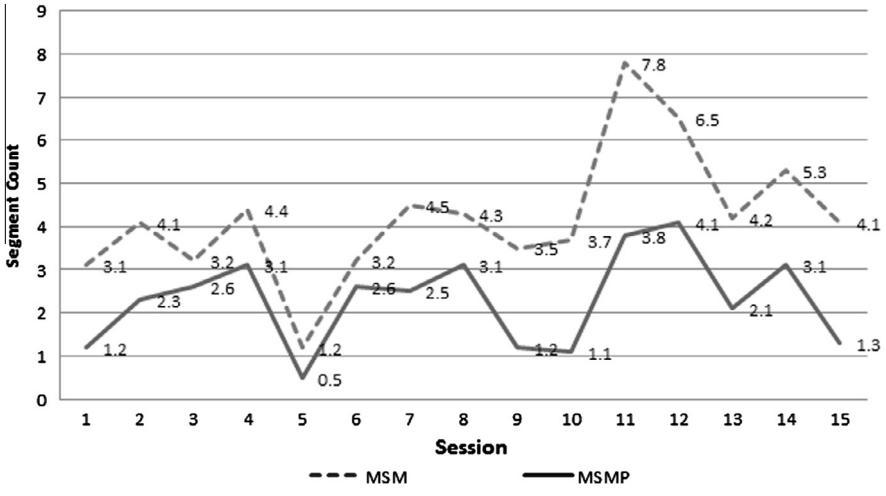

Figure 2 Comparison of MSM and MSMP.

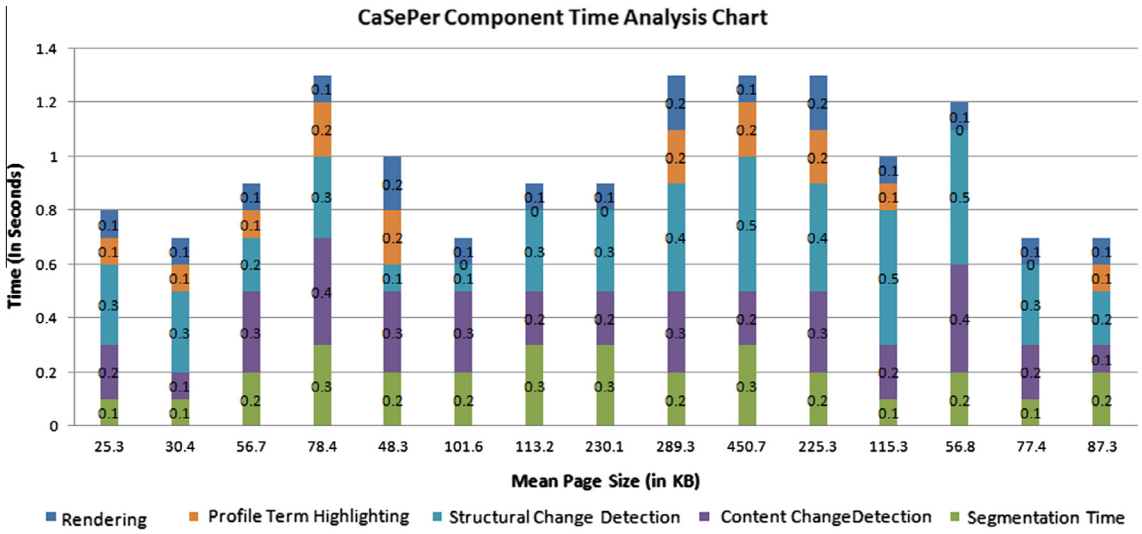

Figure 3 CaSePer component time analysis chart.

varies according to the amount of change that a web page has gone through. In the experiments, the values are measured at ~5% change. The mean of the content and structural change detection time is observed to be 0.24 and 0.31 s. Earlier studies (Wang et al., 2003) have reported a change detection time of ~1 s for documents of ~80 kB size with a 5% change. In the CaSePer model, the mean of change detection, by combining both the structural and content changes, is observed to be 0.55 s, which serves as empirical evidence for the efficiency of the change detection process in the CaSePer model.

5. Summary and conclusions

This paper has proposed a personalized web page change detection model called CaSePer (**Ca**nge detection based on **Se**gmentation with **Per**sonalization). The proposed CaSePer model employs web segmentation followed by edit scripts and node sequence based approaches for change detection in the web pages. The model detects both structural and content changes. The segmentation technique incorporated in the CaSePer model falls under the hybrid segmentation category which encompasses the page tree and densitometry based approaches. To maintain the equity of the segments across various versions of the web page, the model has proposed a "node boundary and cascaded node sequence" based segmentation technique.

The proposed model detects the changes by narrowing down the search space with the help of identified segments. The MD5 hashing technique is utilized for computing the signature of the segments which is further utilized in the change detection process. An additional layer of effectiveness is added to the model with the help of personalization which identifies the user specific changes in the web pages. The task of personalized change detection is achieved with the help of a profile bag that contains the user's interest terms. These interest terms and their linguistic variations are utilized in the personalization process.

The personalized web page change detection model that is based on Segmentation (CaSePer) leads to the following conclusions: Because of the dual-step nature of the proposed model, the complexity associated with the process is reduced by 77.8% in the experiments that were conducted on the prototype built using the CaSePer model. The incorporation of personalization in the change detection model makes the user identify the specific terms of the user's interest. In the experiments that were conducted, the profile-specific changes accounted for 54.77% of the overall changes.

References

Alcic, S., Conrad, S., 2010. Measuring performance of web image context extraction. In: Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD'10, ACM, New York, NY, USA, pp 8:1–8:8.

Alcic, S., Conrad, S., 2011. A clustering-based approach to web image context extraction. In: The Third International Conferences on Advances in Multimedia, MMEDIA, 2011, pp. 74–79.

Artail, H., Fawaz, K., 2008. A fast HTML web page change detection approach based on hashing and reducing the number of similarity computations. Data & Knowledge Engineering 66, 326–337.

Buyukkokten, O., Garcia-Molina, H., Paepcke, A., 2001. Accordion summarization for end-game browsing on PDAs and cellular phones. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, pp. 213–220.

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

Cao, J., Mao, B., Luo, J., 2010. A segmentation method for web page analysis using shrinking and dividing. International Journal of Parallel, Emergent and Distributed Systems 25, 93–104.

Chakrabarti, D., Kumar, R., Punera, K., 2008. A graph-theoretic approach to webpage segmentation. In: Proceeding of the seventeenth International Conference on World Wide Web, ACM, pp. 377–386.

Chawathe, S.S., Garcia-Molina, H. (Eds.), 1997. Meaningful change detection in structured data. ACM SIGMOD, pp. 26–37.

Chawathe, S.S., Rajaraman, A., Garcia-Molina, H., Widom, J. (Eds.), . Change detection in hierarchically structured information. ACM SIGMOD, pp. 493–504.

Cobena, G., Abiteboul, S., Marian, A., 2002. Detecting changes in XML documents. In: Proceedings Eighteenth International Conference On Data Engineering, IEEE, pp. 41–52.

Daniel, S.H., 1997. Algorithms for the longest common subsequences problem. Journal of Association for Computing Machinery 24, 664–675.

Douglis, F., Ball, T., 1996. Tracking and viewing changes on the web. In: Proc. of the 1996 USENIX Technical Conference.

Douglis, F., Ball, T., Chen, Y.-F., Koutsofios, E., 1998. The AT&T internet difference engine: tracking and viewing changes on the web. World Wide Web 1, 27–44.

Fauzi, F., Hong, J.L., Belkhatir, M., 2009. Webpage segmentation for extracting images and their surrounding contextual information. In: Proceedings of the Seventeen ACM International Conference on Multimedia.

Fellbaum, C., 2010. WordNet. In: Poli, R., Healy, M., Kameas, A. (Eds.), Theory and Applications of Ontology: Computer Applications. Springer, Netherlands, pp. 231–243.

Flesca, S., Masciari, E., 2003. Efficient and effective web change detection. Data & Knowledge Engineering 46, 203–224.

Hassanpour, H., Zahmatkesh, F., 2012. An adaptive meta-search engine considering the user's field of interest. Journal of King Saud University – Computer and Information Sciences 24, 71–81.

Jacob, J., Sanka, A., Pandrangi, N., Chakravarthy, S., 2004. WebVigil: An Approach to Just-In-Time Information Propagation in Large Network-Centric Environments. Web Dynamics. Springer, Berlin, pp. 301–318.

Kohlschütter, C., Nejdl, W. (Eds.), 2008. A densitometric approach to web page segmentation. Proceeding of the Seventeenth ACM Conference On Information and Knowledge Management. ACM, pp. 1173–1182.

Kukulenz, D., Reinke, C., Hoeller, N., 2008. Web contents tracking by learning of page grammars. In: Third International Conference On Internet and Web Applications and Services. ICIW'08, pp. 416–425.

Kuppusamy, K.S., Aghila, G., 2013. An ontology based model for user profile building using web page segment evaluation. In: Meghanathan, N., Nagamalai, D., Chaki, N. (Eds.), Advances in Computing and Information Technology, Advances in Intelligent Systems and Computing. Springer, Berlin Heidelberg, pp. 421–430.

La Fontaine, R., 2001. A delta format for XML: identifying changes in XML files and representing the changes in XML. In: XML Europe.

Liu, L., Pu, C., Tang, W. (Eds.), 2000. WebCQ-detecting and delivering information changes on the web. Proceedings of the Ninth International Conference on Information and Knowledge Management. ACM, pp. 512–519.

Miller, G.A., 1995. WordNet: a lexical database for English. Communications of the ACM 38, 39–41.

Pehlivan, Z., Ben-Saad, M., Gançarski, S., 2010. Vi-DIFF: understanding web pages changes. In: Bringas, P.G., Hameurlain, A., Quirchmayr, G. (Eds.), Database and Expert Systems Applications, Lecture Notes in Computer Science. Springer, Berlin Heidelberg, pp. 1–15.

Rivest, R., 2008. The MD5 message-digest algorithm, 1992. RFC1321, Internet Activities Board, Internet Engineering Task Force.

Wang, Y., DeWitt, D.J., Cai, J.Y., 2003. X-diff: an effective change detection algorithm for XML documents. In: Proceedings Nineteenth International Conference On Data Engineering, 2003, pp. 519–530.

Yesilada, Y., 2011. Web page segmentation: a review. [www document]. URL: <http://cng.ncc.metu.edu.tr/content/emine.php> (accessed 21.11.12).