Structural and Visual Comparisons for Web Page Archiving

Marc Teva Law Nicolas Thome Stéphane Gançarski

Matthieu Cord

LIP6, UPMC - Sorbonne University, Paris, France
{Marc.Law, Nicolas.Thome, Stephane.Gancarski, Matthieu.Cord}@lip6.fr

ABSTRACT

In this paper, we propose a Web page archiving system that combines *state-of-the-art* comparison methods based on the source codes of Web pages, with computer vision techniques. To detect whether successive versions of a Web page are similar or not, our system is based on: (1) a combination of structural and visual comparison methods embedded in a statistical discriminative model, (2) a visual similarity measure designed for Web pages that improves change detection, (3) a supervised feature selection method adapted to Web archiving. We train a Support Vector Machine model with vectors of similarity scores between successive versions of pages. The trained model then determines whether two versions, defined by their vector of similarity scores, are similar or not. Experiments on real archives validate our approach.

Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: Digital Libraries

Keywords

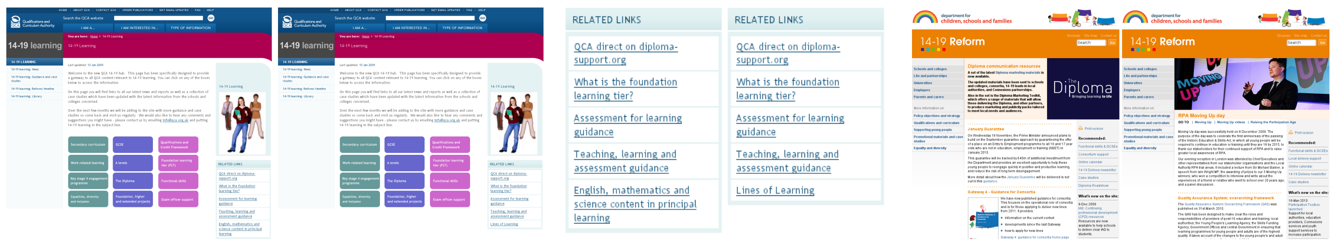
Web archiving, Digital preservation, Change detection algorithms, Pattern recognition, Support vector machines

1. INTRODUCTION

With the explosion of available information on the World Wide Web, archiving the Web is a cultural necessity in preserving knowledge. Most of the time, Web archiving is performed by Web crawlers (bots) that capture Web pages and the associated media (images, videos...). To update archives, crawlers have to regularly revisit pages, but they generally do not know if or when changes appeared. The crawlers cannot constantly revisit a site and download a new version of a page because they usually have limited resources (such as bandwidth, space storage...) with respect to the huge amount of pages to archive. Hence, it is technically impossible to maintain a complete history of all the versions of Web pages of the Web, or even an important part of it. The problem of archivists is then how to optimize crawling so that new versions are captured and/or kept only when changes are important while limiting the loss of useful information. A way to optimize crawling is to estimate the behavior of a site in order to guess when or with which frequency it must be visited, and thus to study the importance of changes between successive versions [1]. For instance, the change of an advertisement link, illustrated in Figures 1(a,b), is not related to the main information shared by the page. In contrast, changes in Figure 1(c) are significant. The crawling of the second version was thus necessary. In this paper, similarity functions for Web page comparison are investigated.

Most archivists only take into account the Web page source code (code string, DOM tree...) [2] and not the visual rendering [3, 4, 1]. However, the code may not be sufficient to describe the content of Web pages, e.g. images are usually defined only by their URL addresses, or scripts may be coded in many different languages that make them hard to compare. Ben Saad et al. [1] propose to use the tree obtained by running the VIPS [3] algorithm on the rendered page. They obtain a rich semantic segmentation into blocks and then estimate a function of the importance of changes between page versions by comparing the different blocks. The VIPS structure of a Web page is a segmentation tree based on its DOM tree. It detects visual structures in the rendering of a Web page (e.g. tables) and tries to keep nodes (blocks) as homogeneous as possible. Two successive paragraphs without html tags will tend to be kept in the same node, whereas table elements with different background colors will be separated in different nodes. Image processing methods have been proposed for Web page segmentation. Cao et al. [5] preprocess the rendering of Web pages by an edge detection algorithm, and iteratively divide zones until all blocks are indivisible. They do not take the source code of Web pages into account. In the context of phishing detection, Fu et al. [6] compute similarities between Web pages using color and spatial visual feature vectors. However, they are only interested in the detection of exact copies.

We propose to investigate in this paper structural and visual features to carry out an efficient page comparison system for Web archiving. We claim that both structural and visual informations are fundamental to get a powerful semantical similarity [7]: structural to catch the dissimilarity if different scripts have the same rendering or if the hyperlinks are changed, visual if the codes of the versions of a Web page are unchanged but a loaded image was updated. Methods combining structural and visual features have been proposed


RELATED LINKS
QCA direct on diploma-support.org
What is the foundation learning tier?
Assessment for learning guidance
Teaching, learning and assessment guidance
English, mathematics and science content in principal learning
RELATED LINKS
QCA direct on diploma-support.org
What is the foundation learning tier?
Assessment for learning guidance
Teaching, learning and assessment guidance
Lines of Learning

(a) Similar versions (b) zoom over the **only** difference between the versions of (a) (c) Dissimilar versions

Figure 1: Similar and dissimilar versions of Web pages. The versions of (a) share the same information, they are exactly similar except the links in (b), they do not need to be crawled twice. The versions of (c) have the same banner and menus but the main information of the page is changed, a second crawling is then necessary.

for content extraction [8], they use the relative positions between elements of pages but no visual appearance features. Additionally, we propose a machine learning framework to set all the similarity parameters and combination weights. We claim to get in this manner a semantic similarity close to archivists' attempts. Our contribution is three-fold: (1) a complete hybrid Web page comparison framework combining computer vision and structural comparison methods, (2) a new measure dedicated to Web archiving that only considers the visible part of pages without scrolling, (3) a machine learning based approach for supervised feature selection to increase prediction accuracy by eliminating noisy features.

2. WEB PAGE COMPARISON SCHEME

Two versions of a given Web page are considered similar if the changes that occurred between them are not important enough to archive both of them. They are dissimilar otherwise (see Figure 1). To compare versions of Web pages, we first extract features from them as described below.

2.1 Visual descriptors

Important changes between page versions will often produce differences between the visual rendering of those versions. We propose to quantify these differences by computing and comparing the visual features in each page version. Each version is described as an image of its rendering capture (snapshot). We compute a visual signature on this captured image for each page. Images are first described by color descriptors, because they seem appropriate for Web page changes and are already used in Phishing Web page detection [6]. We also incorporate powerful edge-based descriptors with SIFT descriptors [9] because they give state-of-the-art performances in real image classification tasks.

For image representation, we follow the well-known Bag of Words (BoW) representation [10, 11]. The vector representation of the rendered Web page is computed based on a sampling of local descriptors, coding and pooling over a visual dictionary. Recent comparisons for image classification point out the outstanding performances of a regular dense sampling [12, 13]. We apply a first strategy called *whole Web page* feature, that samples regularly the visual representation of the whole page. However, the most significant information is certainly not equally distributed over the whole captured Web page. As noted in [14], the most important information is generally located in the visible part of pages without scrolling. A second strategy called *Top of Web page* feature, provides a visual vector using only the features located in the visible part of the page without scrolling.

Since the visible part of a Web page without scrolling depends on the browser window size, we take a generic window height of 1000 pixels, greater than 90% of users' browser resolutions to ensure we do not miss information directly visible by most users. In the next sections, we will denote *the visible part of Web pages without scrolling* by *top of Web pages*.

2.2 Structural descriptors

We extract various features directly from the code of Web pages. For instance, we extract Jaccard indices [2], a similarity value that indicates the preservation between versions of hyperlinks and of URL addresses of images. We assume that similar pages tend to keep the same hyperlinks and images.

We also extract some features from the difference tree returned by the VI-DIFF algorithm [4] that detects some operations between the VIPS structures of versions, e.g. insertions, deletions or updates of VIPS blocks, or even a boolean value returning whether two versions have the same VIPS structure. The more operations are detected, the less similar versions are assumed to be. We denote the features extracted from the VI-DIFF algorithm by VI-DIFF features.

2.3 Similarity between versions

Let V^A be the last archived version of a Web page and V^N the new version of the same Web page. We extract several visual and structural descriptors (see sections 2.1 and 2.2), and use different metrics (Euclidian, χ^2 distances, *etc*) to compare them. Heuristics may be used to set them individually and to select the best similarity function with a manually-tuned threshold to discriminate dissimilar pairs of Web pages from the similar ones.

We propose here an alternate scheme embedding all the similarity functions into a learning framework. Let the M visual feature/metric associations and the N structural similarities be aggregated in a vector $\mathbf{x}$. We can write $\mathbf{x}^T$ as: $[s_v^1(V^A, V^N) \dots s_v^M(V^A, V^N), s_s^1(V^A, V^N) \dots s_s^N(V^A, V^N)]$.

We observed that none of the similarities we experimentally extracted presented a trivial individual decision boundary. However, all of them did seem to follow certain expected patterns, some of them working better than others. Instead of using them individually, we propose to combine those different similarities in a binary classification scheme that returns whether a couple of versions are similar or not by using $\mathbf{x}$, the vector of their similarity scores. Combining both approaches then seems appropriate to have a better understanding of the changes as perceived by human users. Learning combinations of complementary descriptors also makes the categorization task more efficient [15]. We investigate in the next section a statistical learning strategy based on a labeled dataset to classify the vectors $\mathbf{x}$.

3. CLASSIFICATION FRAMEWORK

We are interested in learning distances [16] between versions in a supervised framework to determine whether two versions are similar or not. However, it is not a version classification problem as in many distance learning problems [17]. Indeed, we do not want to classify samples (versions) but similarities. Moreover, our similarities are based on human judgement and allow subtilities as shown in Figure 1.

We then propose to express the learning of the combination of similarities as a binary classification in similarity space: for any couple of versions $(V^A, V^N)_i$, let their class $y_i = 1$ iff V^A and V^N are similar, -1 otherwise. Let $\mathbf{x}_i$ be a vector derived from heterogeneous similarities between V^A and V^N (as defined in subsection 2.3). We train a linear Support Vector Machine (SVM) to determine $\mathbf{w} = \sum_j \alpha_j y_j \mathbf{x}_j$ such that $\langle \mathbf{w}, \mathbf{x}_i \rangle = \sum_j \alpha_j y_j \langle \mathbf{x}_j, \mathbf{x}_i \rangle$ gives us the class of $(V^A, V^N)_i$. The similarity vectors $\mathbf{x}_j$ of training couples $(V^A, V^N)_j$ are used to train an SVM. For any test couple $(V^A, V^N)_i$, the trained SVM returns (1) whether $y_i = 1$ or $y_i = -1$, (2) whether V^A and V^N are similar or dissimilar, (3) whether V^N needs to be archived or not, with V^A already archived. Those three propositions are equivalent.

To study the contributions of the different types of features in the discrimination task, we first train a linear SVM with all the features. Each element w_k of $\mathbf{w}$ corresponds to the weight associated to the k-th similarity feature of $\mathbf{x}$. Therefore, if the learned w_k are close or equal to 0, the k-th similarity features of $\mathbf{x}$ are not determinant for categorization. Such similarities are considered noisy, irrelevant (not discriminant) in determining whether two versions are similar or not. To go one step further, we also propose a more explicit feature selection method based on the automatic *normal based feature selection* [18] that uses the fact that a feature k with the weight w_k close to 0 has a smaller effect on the prediction than features with large absolute values of w_k. Then features with small $|w_k|$ are good candidates for removal. The number of selected features may be set based on data storage and calculation constraints, or iteratively reduced using a validation set.

4. EXPERIMENT RESULTS

4.1 Dataset and settings

We work on a dataset of about 1000 pairs of Web pages manually annotated "similar" or "dissimilar" provided by *The Internet Memory Foundation*[1]. The pages are captured from many different governmental Web sites from the United Kingdom about education, health, sport, justice, industry, security... The identical couples of versions are removed and not taken into account in the evaluation. Finally, 202 pairs of Web pages were extracted: 147 and 55 (72.8% and 27.2%) couples of similar and dissimilar versions, respectively.

To compute visual similarities, we use SIFT and HSV (Hue Saturation Value) color descriptors with visual codebooks of sizes 100 and 200. These are relatively small compared to the sizes used on large image databases but consistent with the size of our base. Bigger codebook sizes did not improve our classification task. The BoWs of page versions are computed using the two strategies described in section 2.1: (1) over the rendering of whole Web pages and (2) the top of Web pages. Euclidian and χ^2 distances are then computed between the BoWs of successive page versions normalized using L^2-norm and L^1-norm, respectively. We also compute for each couple of page versions, the VIPS structures [3] and the VI-DIFF difference trees [4] from which we extract structural similarity values, e.g. the (symmetrized) ratio of identical nodes, boolean values on some criteria such as an identical VIPS structure. In the end, we have 16 visual and 25 structural features.

[1]http://internetmemory.org/

4.2 Binary classification

We use leave-one-out cross-validation (on the 202 pairs) to evaluate our model. We compare our results to the random classifier which automatically predicts the most represented class in the dataset, yielding a baseline accuracy of 72.8%.

Evaluation of visual features.

Selected Visual Features		Accuracy (%)
Whole Web page	Top of Web page	
None	SIFT	84.2
None	color	82.7
None	SIFT + color	**87.1**
SIFT	None	79.7
color	None	80.7
SIFT + color	None	83.2
SIFT + color	SIFT + color	85.1

Table 1: Visual feature classification performances.

We first use only the visual information of pages. Structural similarities of $\mathbf{x}$ are ignored. The accuracies when selecting different subsets of local descriptors (SIFT and color) sampled on whole pages or top of pages are presented in Table 1. SIFT and color descriptors achieve good performances for Web page change detection. Using the top of pages (87.1%) is also a lot more discriminant than using whole pages (83.2%). Combining both of them gives even worse results (85.1%) than using only the top of pages (87.1%). Important changes are more likely to be directly observable whereas changes at the bottom of Web pages, often advertisements, are more likely to be less important and noisy. The accuracies obtained validate our approach.

Evaluation of structural features.

Selected Structural Features		Accuracy (%)
Jaccard Indices	VI-DIFF	
Yes	No	85.1
No	Yes	76.7
Yes	Yes	**87.6**

Table 2: Structural feature classif. performances.

We study in Table 2 the accuracies when different subsets of structural similarities only are used. Jaccard Indices of links are the most discriminant structural features (85.1%) but the other structural features extracted from VI-DIFF are still informative, 4% better than the random classifier.

Structural and visual feature combination evaluation.

We investigate the combination of structural and visual features in Table 3. The accuracy when combining all of them (90.1%) is better than when using only structural (87.6%) or visual (87.1%) features. Visual and structural features are then complementary.

Selected Feature similarities		Acc. (%)
Structural	Visual	
All	All	90.1
All	Top of Web page	92.1
Jaccard indices	All	91.6
Jaccard indices	**Top of Web page**	**93.1**

Table 3: Structural and visual feature classification performances.

Furthermore, we propose to combine in Table 3 the visual and structural features that gave the best accuracies in previous sections. An exhaustive manual selection among all the 41 structural and visual features to find the set that maximizes prediction would be too time-consuming. The accuracy is improved up to 93.1% when combining only Jaccard indices of links and the top of page visual representations.

Concerning misclassified examples, we observed that many dissimilar pairs of versions that were predicted as similar were news pages in which old news were shifted towards the bottom of the page by more recent news. The shifts of these news do not impact the BoW distances since we do not take the spatial information of image patches into account. Many similar pairs of versions predicted as dissimilar had a lot of new irrelevant hyperlinks (significantly more than in Figure 1 (b)). A better detection of important regions and their shifts in position could improve the decision by ignoring their related visual and structural comparisons.

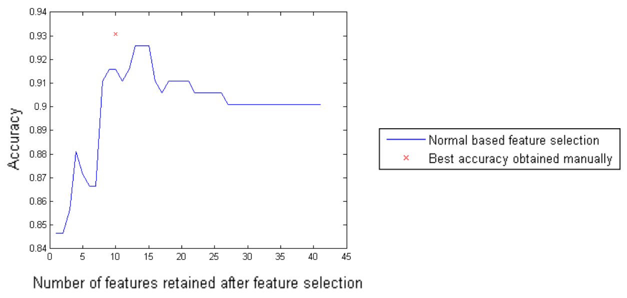


Figure 2: Feature selection performances.

We also investigate the automatic *normal based feature selection* method described in section 3 corresponding to the blue curve in Figure 2. The best accuracy obtained with that automatic method is 92.6% when the 13 to 15 features with the highest absolute values in $\mathbf{w}$ are selected. It is comparable to our best accuracy of 93.1% (Table 3 and red cross of Figure 2) with 10 features selected.

5. CONCLUSION

We have proposed a complete Web page comparison framework effective for Web archiving. We combine structural and visual features to understand the behavior of Web sites and estimate when or with which frequency they must be visited.

We confirm that both structural and visual informations are useful for change detection. We explore several features and similarities. One of the main results is that important changes generally appear at the visible part of Web pages without scrolling. Moreover, we propose a new scheme to learn an optimal similarity combination as a classification problem. Experiments on real Web pages are presented to validate our strategy. A large set of pages with associated labels performed by archivists has been used for a quality evaluation of our visual and structural similarity method.

Acknowledgments. This work was partially supported by the SCAPE Project. The SCAPE Project is co-funded by the European Union under FP7 ICT-2009.4.1.

6. REFERENCES

[1] M. Ben Saad, S. Gançarski, and Z. Pehlivan, “A novel web archiving approach based on visual pages analysis,” in *IWAW 2009*.
[2] M. Oita and P. Senellart, “Deriving dynamics of web pages: A survey,” in *TWAW*, March 2011.
[3] D. Cai, S. Yu, J. Wen, and W. Ma, “Vips: a vision-based page segmentation algorithm,” *Microsoft Technical Report, MSR-TR-2003-79-2003*, 2003.
[4] Z. Pehlivan, M. Ben Saad, and S. Gançarski, “Vi-DIFF: Understanding Web Pages Changes,” in *DEXA 2010*.
[5] J. Cao, B. Mao, and J. Luo, “A segmentation method for web page analysis using shrinking and dividing,” *JPEDS*, vol. 25, 2010.
[6] A.Y. Fu, L. Wenyin, and X. Deng, “Detecting phishing web pages with visual similarity assessment based on earth mover’s distance (emd),” *TDSC*, vol. 3, 2006.
[7] N. Thome, D. Merad, and S. Miguet, “Learning articulated appearance models for tracking humans: A spectral graph matching approach,” *Signal Processing: Image Communication*, vol. 23, no. 10, 2008.
[8] A. Spengler and P. Gallinari, “Document structure meets page layout: Loopy random fields for web news content extraction,” in *DocEng*, 2010.
[9] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, vol. 60, 2004.
[10] W.Y. Ma and B.S. Manjunath, “Netra: A toolbox for navigating large image databases,” in *ICIP 1997*.
[11] J. Fournier, M. Cord, and S. Philipp-Foliguet, “Retin: A content-based image indexing and retrieval system,” *PAA*, vol. 4, no. 2, pp. 153–173, 2001.
[12] S. Avila, N. Thome, M. Cord, E. Valle, and A. Araújo, “Bossa: Extended bow formalism for image classification,” in *ICIP 2011*.
[13] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, “The devil is in the details: an evaluation of recent feature encoding methods,” *BMVC*, 2011.
[14] R. Song, H. Liu, J.R. Wen, and W.Y. Ma, “Learning block importance models for web pages,” in *WWW 2004*.
[15] D. Picard, N. Thome, and M. Cord, “An efficient system for combining complementary kernels in complex visual categorization tasks,” in *ICIP 2010*.
[16] L. Yang and R. Jin, “Distance metric learning: A comprehensive survey,” *Michigan State University*, pp. 1–51, 2006.
[17] A. Frome, Y. Singer, and J. Malik, “Image retrieval and classification using local distance functions,” in *NIPS 2006*.
[18] D. Mladenić, J. Brank, M. Grobelnik, and N. Milic-Frayling, “Feature selection using linear classifier weights: interaction with classification models,” in *SIGIR 2004*.