

DETECTING WEBSITE REDESIGNS VIA TEMPLATE SIMILARITY ON STREAMS OF DOCUMENTS

Thomas Gottron

Institut für Informatik, Johannes Gutenberg-Universität, Mainz, Germany
gottron@uni-mainz.de

ABSTRACT

Most websites undergo a redesign from time to time. Along with the change of the appearance of the site comes a different document structure. Hence, redesigns can be detected by observing changes in the structural similarity of monitored HTML documents. Assuming further to monitor not a fixed document set but a series of the newest documents (e.g. provided by an RSS feed) transforms the task of redesign detection into a particular change detection operation on streams of documents. This paper describes and evaluates a simple and three more elaborated approaches to the problem. We show that the detection of redesigns can be achieved automatically, effective and efficient.

KEYWORDS

Redesign Detection, Change Detection, Template Similarity, Document Streams

1. INTRODUCTION

From time to time, most websites on the World Wide Web change their layout. The reason for such a redesign might be to improve usability, to include new functionality, or simply to provide a new, more up-to-date look and feel. Often also an exchange of the employed Web Content Management System (WCMS) might issue the need of a redesign. In any case, a redesign causes a change in the structure of the documents of the website, because they are based on a new template to reflect the changed design.

Accordingly, programs which rely on certain structural characteristics of the documents, e.g. to extract information or detect changes in certain contents, might eventually run into problems. Information extraction programs might fail entirely or extract erroneous or incomplete data. Change detection is affected by a redesign in a different way. Very likely, the algorithms developed for change detection report a change of a monitored document after a redesign. Modifications of the DOM structure and new template induced texts cause the documents to appear changed. The actual main content, instead, might still be the same as before the redesign. All these applications might benefit from an automatic detection of website redesigns. It would allow a human user to check their behaviour or even to take corrective measures automatically.

The automatic detection of a website redesign is of interest for other applications as well. Template detection systems or applications involving content extraction can be triggered to detect new template structures or to revise their extraction strategy. Search engines, might use a detected redesign as a motivation to issue a new crawl of the web site in order to update their index with new documents or changes in the link structure. Finally, also human users might wish to be notified about a redesign, even if simply for matters of interest or curiosity.

In this paper we describe four different methods to detect website redesigns. The algorithms are all based on an efficient calculation of template similarity measures for HTML documents. By observing a stream of the newest or lately updated documents, the idea is to maintain a history

of the similarity of the templates of the last seen documents. The algorithms use this history to detect major shifts in template similarity and, thereby, a website redesign.

We proceed with a short look at related work in the next section, before describing our template similarity measure in section 3. Afterwards we explain the concept of document streams (4.1), how we maintain a history of recently seen documents (4.2) and describe our detection algorithms (4.3). After evaluating the different approaches in 5 we summarise the results and take a look at possible directions for future work.

2. RELATED WORK

To our knowledge, the direct and automatic detection of website redesigns has never been addressed. The general detection of changes in web documents, though, is an area in which a lot of research has been done. Systems like WebCQ [1], the change detection mechanism in CMW [2] or WebVigil [3] typically focus on detecting changes in the content of a document. Beyond research projects, there are also several publicly available online Web monitors (www.detectchange.com, www.changealarm.com, www.changedetection.com) allowing to track changes in web documents. A redesign of the host website, however, will in most cases be considered a change in the content – even if the actual content of the document remained the same. While Artail and Fawaz in their work [4] concentrate on detecting content changes fast and efficient, they mention the discovery of layout changes as a task which is of interest as well.

Wrapper verification deals with the impact of a redesign on information extraction scenarios. However, it addresses merely the effects but does not directly analyse and detect the cause of the problem. Kushmerick [5] decides whether a wrapper is still functional based on the data it provides. If the extracted information does not fit into a learned pattern, the conclusion is that the website has been redesigned.

The structural comparison of web documents has been dealt with in several papers. Cruz et al. describe different distance measures for web documents in [6]. Buttler [7] introduced DOM tree path shingling making use of the shingling technique of Broder et al. in [8]. In [9] we compared several existing and new document distance measures and found out that some simple measures are actually most suitable to detect the changes and differences caused by template structures. The common tag sequence shingle (CTSS) measure we developed in this context can be computed efficiently and, regarding the separation of documents based on different templates, performed best compared to all other approaches.

3. TEMPLATE SIMILARITY

A template similarity measure is a function to compare two HTML documents and analyse their structural similarity. The result of this analysis is a real value in the interval $[0,1]$, where a value close to 1 represents a high similarity and a value close to 0 represents a low or no similarity. The idea, that a template framework of a WCMS strongly influences the structure of the managed documents leads to the conclusion, that a high structural similarity value corresponds to the documents being based on the same template.

To compare documents we use CTSS from [9]. Due to space limitations we restrict ourselves here to a formal definition and refer to the original publication for details. CTSS splits the sequence of tags in a document D into a set of shingles $ts(D)$ (a shingle is a continuous subsequence of fixed length). The CTSS measure uses the overlap between the tag shingle sets of two different documents to determine similarity:

$$sim(D_1, D_2) = \frac{|ts(D_1) \cap ts(D_2)|}{\max(|ts(D_1)|, |ts(D_2)|)}$$

The effect of a website redesign on the template similarity can be seen quite impressively in figure 1. It graphically represents the similarity matrix of a collection of about 140 documents taken all from the same website. Each pixel represents the comparison of two documents, i.e. the pixel in column i from the left and row j from the top represents the similarity value between document D_i and D_j . The higher the similarity, the brighter is the colour of the pixel. The monitored website had been redesigned after about two thirds of the documents were downloaded. As documents in the matrix are ordered by their time of publication, the change to a new template causes the matrix to show two submatrices of high similarity (the brighter squares) and two of low similarity (the darker rectangles).

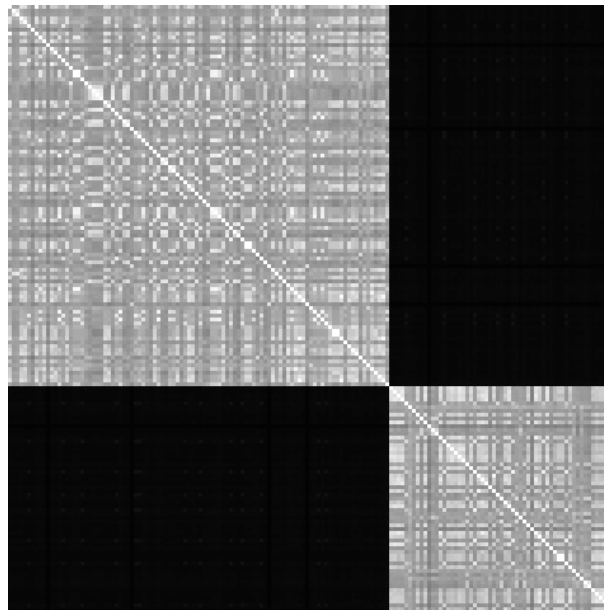


Figure 1. Template similarity matrix for 140 documents taken from www.welt.de. The redesign of this website can be recognised by the clear subdivision into brighter and darker areas.

4. REDESIGN DETECTION

The above example suggests that template similarity measures are a suitable approach to detect website redesigns. However, it is too time consuming to compute a full similarity matrix for all documents ever published on a web site. Therefore we will take a slightly different approach. We assume the documents of a web site to be available as a stream of documents and monitor only similarity changes in recent documents.

4.1. Streams of Documents

To notice structural changes of the documents of a website it is necessary to observe the documents frequently. We model these frequent observations over a stream of documents. This notion corresponds in the easiest case to follow the latest entries in an RSS feed. If such a syndication feed is not available, we can assume new or constantly monitored documents as the output of a crawling process to be fed into a document stream.

Formally, we consider a series of HTML documents d_i , $i = 0, 1, 2, 3, \dots$ which are published at time $t(d_i)$, with $t(d_i) \leq t(d_{i+1})$ as a stream of documents.

4.2. Document History

As we do not want to consider the similarity of all documents we have ever seen, we maintain only a limited history of the last h seen documents. For a new document d_i we compute a vector S_i of similarities with the documents in the history:

$$S_i = \begin{pmatrix} \text{sim}(d_i, d_{i-1}) \\ \text{sim}(d_i, d_{i-2}) \\ \vdots \\ \text{sim}(d_i, d_{i-h}) \end{pmatrix}$$

Document d_i is then added to the set of history documents itself and replaces the oldest document in the history, namely d_{i-h} . Note, that we do not need to keep a copy of the full document in the history, its shingle set $ts(D)$ is sufficient to calculate further similarity values.

So, instead of computing the complete similarity matrix over all documents, we effectively compute only a stripe of width h in the matrix, which lies below the main diagonal. For the matrix of our example above this stripe and its construction are shown in figure 2. Again, the redesign can be recognised quite well by the dark triangle in the stripe.

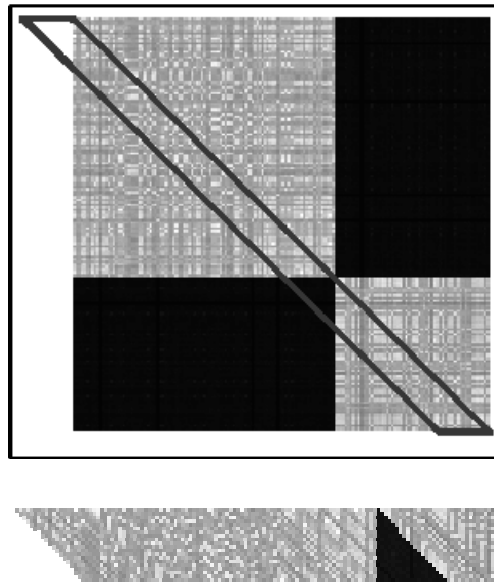


Figure 2. Graphical construction of the similarity stripe we calculate. The stripe is shown for a history of 20 documents. The “empty” triangle to the left represents the beginning of the stream when we do not have any documents in the history.

There is no need to keep the full stripe stored in memory. It is sufficient to remember the part of it which is still relevant for the documents in the history. After adding document d_i , this corresponds to a matrix S of similarity vectors:

$$S = (S_{i-h+1} \quad S_{i-h+2} \quad \cdots \quad S_i)$$

Figuratively speaking, we only need to keep track of a small, square-shaped view on the similarity stripe. The matrix S and its representation in the stripe are shown in figure 3. Note, that the matrix S – though it contains similarity values – does not have the typical symmetric shape of a classical similarity matrix. As the set of history documents changes continuously,

older documents at their time were compared to a different set of documents than the newer ones. Effectively, after adding the similarity vector of the most recent document d_i , the entry (s_{jk}) in our matrix represents the value $sim(d_{i-h+k}, d_{i-h+k-j})$.

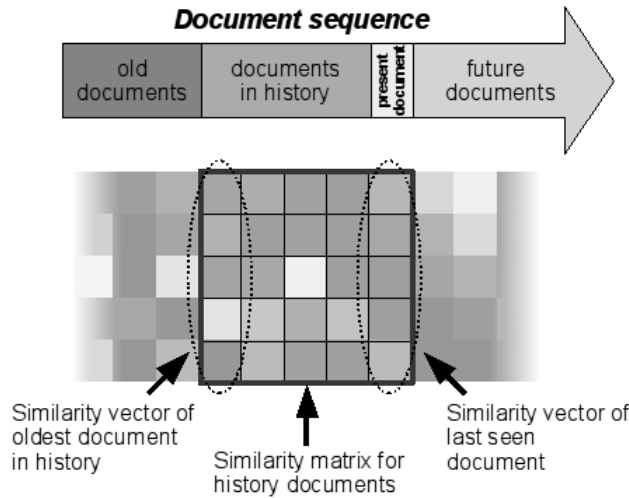


Figure 3. Instead of storing the entire similarity stripe, we only need to consider the matrix of similarity vectors for the documents in the history set.

The reason why we maintain a history of several documents (instead of simply comparing the current with the previous document) are outliers. The document stream might contain documents which are based on different templates. They occur because of references to external sites, particular and exceptional pages, or due to the use of different templates within the same website. The similarity stripe in figure 4 illustrates this problem. This stripe actually shows a lot of outliers (represented by the dark lines of low similarity), while no redesign occurred during the observation of these documents.



Figure 4. Similarity values in a document stream with template outliers.

4.3. Detection Algorithms

To detect a redesign in the stream of documents – or more precise in the documents we have in our history – we need to identify a permanent shift from one document template to another one. More formal again, we want to determine a document d_i , for which all documents d_j with $j < i$ are based on a different template, while the template of document d_i occurs again frequently for documents appearing later on in the stream. To do so, we propose four different approaches:

NEW-AVG-DROP: The simplest idea to detect website redesigns is to compare the average similarity of each new document with the one of the history documents. This is, for d_i we calculate $avgsim(d_i) = \frac{1}{h} \sum_{j=i-h}^{i-1} sim(d_i, d_j)$. If this average similarity of a new document drops significantly, we have seen a new template. However, a single document based on a different template might not indicate a redesign, but might simply be due to an outlier. Hence, this approach might raise a lot of “false alarms”, i.e. report a redesign when non occurred.

MAX-AVG-DIFF: Because of outliers we have to see several documents based on a different template before making our decision. The redesign in figure 2 was reflected by a dark triangle in

the stripe representation. In this and the next approach we try to discover this triangle structure. Note, that in this way we detect a redesign exactly h documents after it actually happened, as otherwise the triangle is not yet completely formed.

To identify the redesign, we subdivide the matrix S into a lower left and upper right triangle as shown in figure 5. The lower triangle, denoted in the figure by the label (1), corresponds to comparisons between the documents currently in the history and others seen still earlier. The values in the upper triangle, labelled with (2), correspond to similarities between the documents currently in the history set. Hence, low similarity values in the lower triangle and high values in the upper triangle indicate a change of template at document d_{i-h} . The borderline between the two triangles corresponds to the shift from the old template to a new one.

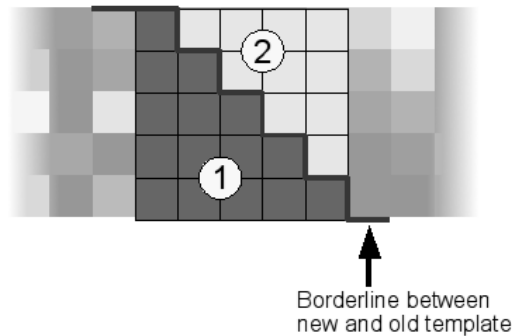


Figure 5. Subdivision of history similarity matrix into triangles. After a redesign, the lower left triangle (1) contains very low similarity values, while the upper right triangle (2) contains high similarity values.

The MAX-AVG-DIFF approach detects the low and high values in the two triangles in a rather simple way. We compute the maximum similarity of the lower triangle to find the most similar document pair, of which one was seen before document d_{i-h} and one afterwards. In the upper triangle we compute the average similarity between history documents. By using the average we compensate the impact of outliers occurring shortly after the transition from the old template to the new one. We compare these two values and if the maximum in the lower triangle is less than the average of the upper one we consider this a hint for a redesign.

STATISTICS: Instead of the heuristic of comparing the maximum with the average similarity to detect the differences in the triangles, we can also employ statistics. Therefore, we analyse the estimates for the mean and standard deviation of the similarity values in each triangle. A statistical t-test with a significance value of 99.9% then serves to identify a strong shift in the distributions of the similarity values in the two triangles.

CLUSTER: The last approach we discuss in this paper does not identify the triangle structure in the similarity stripe, but is based on a continuously updated template cluster analysis of the history documents. For each template we form a cluster of documents that are based on this template. A new document is then assigned to the cluster of its own template. This can be achieved by determining the maximum CTSS similarity with the documents in the clusters. A new template is detected if the maximum similarity to all other documents is below a threshold t , which we chose according to [9] to be 0.15. In this case we create a new cluster.

A redesign corresponds to a permanent shift from an existing to a new cluster. Such a permanent shift can be detected by introducing a second horizon $h_2 < h$. We compare the number of document in each cluster before and after h_2 . If we find a cluster which contains only documents before h_2 , while another cluster contains only documents after h_2 , we have found a

permanent shift and hence a redesign. Compared to the two previous approaches, we detect the redesign also earlier, namely h_2 documents after it occurred. To avoid a redesign detection because of clusters formed by single outliers, we demand the shift to take place between large clusters, i.e. clusters which contain a minimum rate of all observed documents.

5. EVALUATION

We evaluate our four algorithms on different series of documents. We will then compare their performance in detecting website redesigns, but also their tendency to raise “false alarms”, i.e. report a redesign while actually none occurred.

5.1. Data

For evaluation we use real data, i.e. documents taken from different websites. The documents are combined into timely ordered series to simulate the document stream behaviour. Further, we created different series to reflect several possible scenarios of interest:

- The simplest scenario are websites with a homogeneous layout for all documents and no outliers. Further, in these series no redesign occurs. Hence, our algorithms should process the documents without any notification.
- The second scenario is more difficult. The series still do not contain a redesign event, but suffer to several degrees from outliers and inhomogeneous layouts.
- The third scenario is based again on documents with a homogeneous template, but this time with a redesign occurring. In this category we actually use one series of documents, where a redesign occurred during the collection of documents. In a certain way, this series (taken from *www.welt.de*) represents the most realistic scenario. The second series was created by combining old and new documents taken from another website (*www.heise.de*), but without the documents which were published right before or after the transition to the new layout. Finally, lacking other suitable documents, we combined the homogeneous documents from two websites into one new, artificial series. However, the similarity values appearing in this sequence do not seem to be too distorted, when comparing them with the ones of the other two series.
- The last scenario corresponds to the missing combination of a document series with outliers and a redesign. This is the hardest case for redesign detection. Unfortunately we did only have one such series from a single website (*www.spiegel.de*), and again we are missing the documents right before and after the redesign. The other two series were created artificially by combining documents from two other very inhomogeneous series as already described above.

The data of all our evaluation series is shown in detail in table 1. It lists the names of the series (which correspond to the websites the documents were taken from), the number of documents involved and shows the similarity stripes for a horizon of 20 documents. The similarity stripes allow to get a quick impression of whether the data is homogeneous or contains outliers (recognisable again by the darker lines). For the series which do contain redesign events we manually analysed the documents to determine at which point the redesign occurred. In this way we can compare the results of our algorithms with a ground truth.

5.2. Results

We fed the four redesign detection algorithms with each of the document series described above. Each algorithm was supposed to report the respectively first document it considers to

have been published after a redesign. Further, to see the influence of the amount of documents kept in the history, we ran each experiment with history setting of 5, 10 and 20 documents. For the CLUSTER algorithm we set the h_2 horizon to 4 documents.

Comparing the documents found by the algorithms with the ones we manually determined to be the first after a redesign, we can construct confusion matrices for our algorithms. Table 2 lists the number of correctly identified redesigns (TP), documents correctly identified to continue to use an old design (TN), false alarms of a redesign (FP) and missed redesigns (FN).

Looking at the result, the CLUSTER and the MAX-AVG-DIFF algorithms seem to be the most suitable for detecting redesigns. With a history of 20 documents, they make only three and four mistakes respectively. NEW-AVG-DROP, as expected, tends to report a redesign for each outlier. The STATISTICS approach, instead, is surprisingly disappointing and also the only algorithm which does not improve performance when increasing the size of the history.

Table 1. Evaluation data for redesign detection, including the similarity stripes for a horizon of 20 documents.

Data series	Docs
Homogeneous templates; no redesign	
Herald	299
Golem	329
Yahoo	226
Inhomogeneous templates and outliers; no redesign	
FAZ	309
BBC	345
Tagesschau	189
Homogeneous templates; redesign	
Welt	134
Heise	631
Focus-Chip	484
Inhomogeneous templates and outliers; redesign	
Spiegel	441
FAZ-Tagesschau	499
Repubblica-ZDF	370

Table 2. Performance of the four redesign detection algorithms.

Algorithm	Horizon	TP	TN	FP	FN
NEW-AVG-DROP	5	4	4263	222	2
	10	4	4308	177	2
	20	6	4327	158	0
MAX-AVG-DIFF	5	6	4352	133	0
	10	6	4480	5	0
	20	6	4481	4	0
STATISTICS	5	4	4316	169	2
	10	5	4329	159	1
	20	5	4268	217	1
CLUSTER (h_2 horizon set to 4)	5	4	4475	10	2
	10	4	4482	3	2
	20	6	4482	3	0

In general, regarding the size of the history set we observe, that except for STATISTICS all algorithms benefit from a larger history. In general, a history of 20 documents seems to be sufficient for reliable redesign detections.

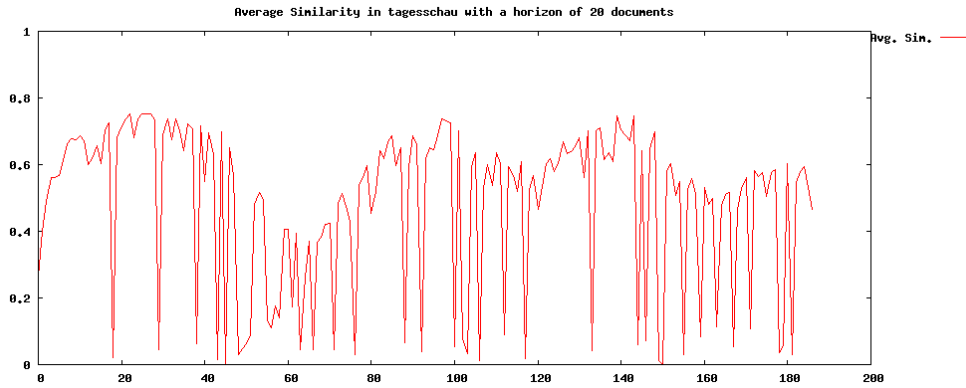


Figure 6. Average similarity of a newly seen document. Outlier documents cause sudden drops of the value, even though no redesign occurred.

Going more into the details, the graph in figure 6 illustrates the problem of the NEW-AVG-DROP algorithm. Each document which is based on a template different from the previous ones causes a significant drop in the average similarity of the new document. Hence, it has extreme problems with document series containing outliers or different templates.

Also the STATISTICS reports far too many “false alarms”. Even though we used very strict confidence criteria for the statistical test, a few outliers in the lower triangle are sufficient to shift the distribution of similarity values enough for considering it caused by a redesign.

The MAX-AVG-DIFF algorithm instead performs outstanding. Figure 7 shows a plot of the two indicators involved in this approach: the average similarity in the upper triangle and the maximum similarity in the lower triangle. A redesign causes a clear peak in the lower triangle, which can easily be detected.

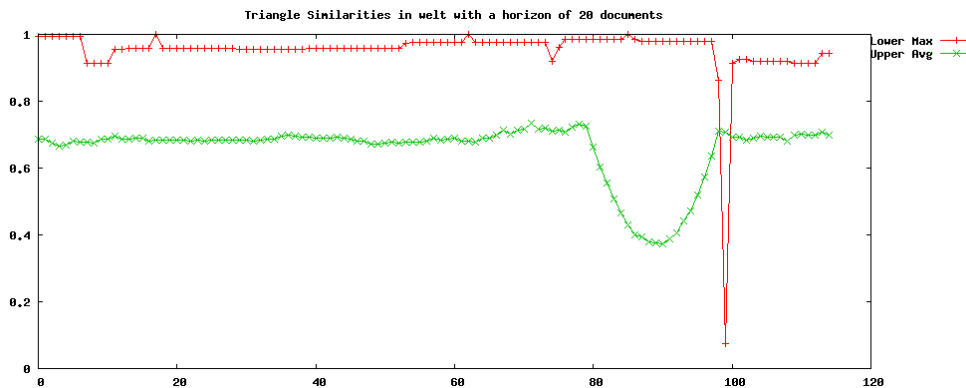


Figure 7. The indicators used in the MAX-AVG-DIFF algorithm. Redesigns have a clear peak in the maximum similarity value of the lower triangle.

The four false reports of a redesign occurred in two of the artificial test series (Focus-Chip and FAZ-Tagesschau) and were all preceding immediately the actual redesign. The reason in the Focus-Chip series was that the general similarity of documents before the simulated redesign was significantly lower than with the new template. In the case of the FAZ-Tagesschau series,

the reason were two outlier documents exactly at the transition point from the old design to the new one (such a peculiar case would actually qualify also according to our formal definition as three redesign events in a row).

The CLUSTER approach is improving the qualitative results of the MAX-AVG-DIFF method only marginally. Its biggest advantage is, that it can report the redesigns earlier, i.e. need to see far less documents based on the new template, before it can make its decision. Hence we can say, that CLUSTER is the most suitable approach for redesign detection.

6. SUMMARY AND OUTLOOK

Detecting website redesigns is an interesting topic for several applications, such as change detection, information extraction, web mining and content extraction. To our knowledge it has not been addressed directly before. We presented four different approaches to detect redesigns by observing a stream of documents. By analysing a similarity matrix of the last seen documents our MAX-AVG-DIFF and the CLUSTER algorithms are detecting website redesigns reliably. Both methods do not get confused by outliers or variations of different templates within the same website. The CLUSTER algorithm has the advantage to detect redesign faster, i.e. after less documents based on the new design.

An improvement can be achieved by reducing the amount of documents we need to see after a redesign to actually discover the change of template. However, there will always be a tradeoff between being able to handle outliers (especially several in a row) and a fast redesign detection. A conceptual improvement would be to incorporate a measure for the certainty for having discovered a redesign. In this way the detection algorithm could notify other programs earlier, but also tell them it is not very sure yet about the discovered change.

REFERENCES

- [1] Liu, L., Pu, C. & Tang, W. (2000) "WebCQ - detecting and delivering information changes on the web", *CIKM '00: Proceedings of the 9th international conference on Information and knowledge management*, pp512-519.
- [2] Flesca, S. & Masciari, E. (2003) "Efficient and effective web change detection" *Data & Knowledge Engineering*, Vol. 46, No. 2, pp203-224.
- [3] Chakravarthy, S., Jacob, J., Pandrangi, N. & Sanka, A. (2002) "Webvigil: An approach to just-in-time information propagation in large network-centric environments", *WebDyn '02: 2nd International Workshop on Web Dynamics*, pp301-318.
- [4] 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*, Vol. 66, No. (2), pp326-337.
- [5] Kushmerick, N (2000) "Wrapper verification", *World Wide Web*, Vol. 3, No. 2, pp79-94.
- [6] Cruz, I. F., Borisov, S., Marks, M. A. & Webbs, T. R. (1998) "Measuring structural similarity among web documents: preliminary results", *EP '98: Proceedings of the 7th international Conference on Electronic Publishing, Artistic Imaging, and Digital Typography*, pp513-524.
- [7] Buttler, D. (2004) "A short survey of document structure similarity algorithms", *IC '04: Proceedings of the International Conference on Internet Computing*, pp3-9.
- [8] Broder, A. Z., Glassman, S. C., Manasse, M. S. & Zweig, G. (1997) "Syntactic Clustering of the Web", *Computer Networks*, Vol. 29, No. 8-13, pp1157-1166.
- [9] Gottron, T. (2008) "Clustering template based web documents", *ECIR '08: Proceedings of the 30th European Conference on Information Retrieval*, pp40-51.