Improving the Quality of Web Archives through the Importance of Changes[*]

Myriam Ben Saad and Stéphane Gançarski

LIP6, University P. and M. Curie,
4 place Jussieu 75005, Paris, France
{myriam.ben-saad,stephane.gancarski}@lip6.fr

Abstract. Due to the growing importance of the Web, several archiving institutes (national libraries, Internet Archive, etc.) are harvesting sites to preserve (a part of) the Web for future generations. A major issue encountered by archivists is to preserve the quality of web archives. One way of assessing the quality of an archive is to quantify its completeness and the coherence of its page versions. Due to the large number of pages to be captured and the limitations of resources (storage space, bandwidth, etc.), it is impossible to have a complete archive (containing all the versions of all the pages). Also it is impossible to assure the coherence of all captured versions because pages are changing very frequently during the crawl of a site. Nonetheless, it is possible to maximize the quality of archives by adjusting web crawlers strategy. Our idea for that is (i) to improve the completeness of the archive by downloading the most *important* versions and (ii) to keep the most important versions as coherent as possible. Moreover, we introduce a pattern model which describes the behavior of the importance of pages changes over time. Based on patterns, we propose a crawl strategy to improve both the completeness and the coherence of web archives. Experiments based on real patterns show the usefulness and the effectiveness of our approach.

Keywords: Web Archiving, Data Quality, Change Importance, Pattern.

1 Motivation

The main goal of web archiving institutes (national libraries, Internet Archive, etc.) is to preserve the history of web sites for future generations. Most often, web archiving is automatically performed using web crawlers. Web crawlers visit web pages to be archived and build a snapshot and/or index of web pages. In order to maintain the archive up-to-date and to preserve its quality, crawlers must revisit periodically the pages and update the archive with fresh images. We define the quality of an archive by its completeness and by the coherence of its page versions. Completeness measures the ability of the archive to contain

[*] This research is supported by the French National Research Agency ANR in the CARTEC Project (ANR-07-MDCO-016).

A. Hameurlain et al. (Eds.): DEXA 2011, Part I, LNCS 6860, pp. 394–409, 2011.

the largest amount of useful versions. Coherence measures how much the archive reflects the snapshot of web sites at different points in time. An ideal approach to preserve the quality of archives is to crawl all pages of a site at the same time at every modification or to prevent pages content from changing during the crawl. Of course, this is practically infeasible because web sites are autonomous and thus out of control. In fact, it is impossible to maintain a complete archive of the whole web site (*i.e.* containing all the versions of all the pages) because web sites are evolving over time and allocated resources are usually limited (such as bandwidth, storage space, site politeness rules). Also, crawling a large web site may span hours and even days. This increases the risk of page changes during the crawl that leads to incoherence between archived pages. Page versions (of a same site) are considered incoherent, if they have never existed together at any point in time in the real web history.

Though it is impossible to preserve a perfect quality of web archives, this quality can be improved if web sites are crawled "at the right moment". Our work aims to adjust the crawling strategy so that the built archive will be as complete and as coherent as possible. Our ideas, for that, is (i) to improve the completeness of the archive by downloading the most *important* versions and (ii) to assure that coherent versions we obtain, are the most important ones. An important version is a version that has important change with respect to the last one archived of the same page. Hence, unimportant changes in the page (*e.g.* advertisements, decoration, etc.) can be ignored and useful information is captured by a single crawl, maximizing the use of resources. Up to now, most crawling strategies [5,13,10,16] do not consider the *importance of changes* that have occurred between versions. They consider the crawl useful even if the captured version is almost equal to the previous one. Moreover, they estimate the frequency of page changes based on the homogeneous Poisson model [8,6] with a constant change rate λ. This model is valid when the time granularity of changes is longer than one month as shown in [15] which is far from being the common case. For instance, our work is applied on a repository for the French National Audiovisual Institute (INA) which creates a legal deposit to preserve French radios and televisions web pages and related pages. Those web pages, such as on-line newspapers change very frequently (more than once a day). As the time granularity, for those pages, is much shorter than one month, the homogeneous Poisson model is not valid as demonstrated in [15].

We have the idea to use page change patterns. A pattern models the behavior of the importance of the changes over time, during for example a day. Based on patterns, the evolution of changes can be accurately predicted over periods of time and exploited to optimize crawlers. In previous work [3], we have monitored French TV channels pages (*France Télévision*) over a period of one month. Each page was hourly crawled every day. Then, we have discovered patterns by using a statistical summarization technique. Based on these patterns, we show, in this paper, how the strategy of crawlers can be adjusted to improve quality of archives. As far as we know, the concepts of changes importance and patterns had never been exploited to optimize the quality of archives. Related crawl policies that have considered the importance/relevance of pages are mostly based on the PageRank (also similarity to keywords of queries) but the importance of changes between versions have been ignored so far. Moreover, this work is the first to address both completeness and coherence, at same time, in the context of web archiving. The main contributions of this paper can be summarized as follows:

- A description of our archiving model based on two concepts: changes importance and page changes pattern.
- A definition of two quality measures (completeness and coherence) to assess the quality of the archive. These measures consider the importance of archived versions.
- A novel crawl strategy based on patterns that uses the importance of changes to improve both the completeness and the coherence of web archives.
- An implementation of our strategy and experimental results that demonstrate its effectiveness.

This paper is structured as follows. In Section 2, related works are presented and discussed. Section 3 introduces the different concepts used in the paper. Section 4 describes our web archiving model. Section 5 defines two measures, completeness and coherence that assess the quality of archives. Section 6 describes our pattern-based strategy using the importance of changes. Section 7 discusses experimental results. Section 8 concludes.

2 Related Works

In recent years, several projects have addressed issues involved by web archiving. An overview of these main issues is given by Masanès [12]. Many studies are closely related to our work, in the sense that they aim at optimizing crawlers. Brewington and Cybenko [4] estimate how often web sites must be re-indexed based on the analysis of page changes for more fresher indexes. Pandey and Olston [13] propose a recrawl scheduling strategy based on information longevity to improve the freshness of web pages. In [8], Cho et al. estimate the frequency of page changes based on the Poisson process. In other studies [6,7], they propose efficient policies to improve the freshness of web pages. In [9], they propose a crawl strategy to download the most important pages first based on different metrics (*e.g* similarity between pages and queries, rank of a page, etc.). The research of Castillo et al. [5] goes in same direction. They propose a crawl strategy that retrieves the best ranked pages. Most designed strategies that have considered the importance of pages are based on the PageRank (also similarity to keywords of queries) but do not take into account the importance of changes between versions. Moreover, existing crawl strategies are mostly based on change rate estimated by the Poisson Process. As already mentioned in [15], researchers demonstrate that the Poisson process is not valid for pages changing several times per day as it is the case in our context. Similarly to our work, Adar et al. [1] propose models and analysis to characterize the amount of change on the web at finer grain (frequent updates per day). However, they do not propose a method to estimate the importance of (structural and content) changes detected between pages versions. According to them, their change analysis can be used to refine crawler policies but, no effective strategy has been proposed.

Recent studies address the issue of improving the quality of archives. Spaniol et al. [16] propose a crawling strategy in order to optimize the coherence of web sites captures. In [17], they present visualization strategies to help the archivist at understanding the nature of coherence defects. In another study [10], they define two quality measures (blur and sharp) to assess the quality of the archive and propose a framework, coined SHARC, to optimize site-capturing policies. The two approaches proposed in [10,16] to improve the coherence and the sharpness of the archive are based on multiple revisits of web pages. However, in our work, we assume that web crawlers have limited resources which prevent from revisiting a page too often. Also, the importance of changes between page versions has been ignored.

Our work is also related to pattern mining area. Patterns are widely introduced and implemented for different applications such as trajectories of objects, weather, DNA sequences, stock market analysis, etc. They were exploited to detect anomalies, to predict data behavior (or trend), or more generally, to simplify data processing, etc. It is impossible to give here a complete coverage on this topic but interested readers can refer to [11] for example. A large coverage of pattern mining approaches is given. To the best of our knowledge, patterns have never been used to improve web archiving. In [3], we presented, through a case study, steps and methods to discover patterns from French TV channels pages. Here, we investigate how these patterns can be used to improve completeness and coherence of web archives based on the importance of changes.

3 Concepts

In order to better understand the next sections, we introduce here the different concepts that we use in this paper.

- **Quality of an archive.** We evaluate the quality of the archive through the two following measures.

  - ***Completeness.*** It measures the ability of the archive to contain the largest amount of useful page versions. This quality measure is relevant because it is very frequent, while navigating through the archive, that users cannot reach some web pages. Those missed pages had not been downloaded (at right moment) before they disappear from the web.

  - ***Coherence.*** It measures the ability of the archive to reflect the states (or snapshots) of a web site at different points in time. Indeed, when users navigate through the archive, they may want to browse (part of) sites instead of individual pages. Coherence ensures that if users reach a page version of a site, they can also reach other pages versions of the same site corresponding to the same point in time.

  In the rest of the paper, we use the term quality to express both completeness and coherence of the archive.

- **Importance of a version**

  An important version is a version of an important page that has significant changes compared with the last version archived of the same page. Therefore, the importance of a version depends on:

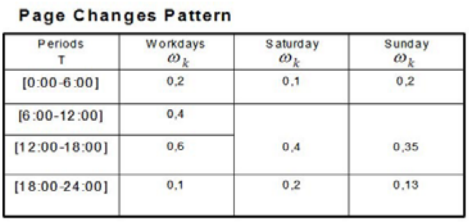
  1. the importance of the corresponding page (*e.g.* PageRank, similarity to keywords of a query, etc.)
  2. the importance of the changes that have occurred on the page since the last version archived.

  Changes between two page versions are detected by the Vi-DIFF algorithm [14]. First, Vi-DIFF extends a visual segmentation algorithm to partition the web page into multiple blocks. Blocks simulate how a user understands the page layout structure based on his visual perception. Then, Vi-DIFF detects *structural* changes (*i.e.* an insert, a move, etc. at level of blocks composing the page) and *content* changes (*i.e.* a delete, an update, etc. at level of texts, hyperlinks and images inside blocks). The importance of changes between two versions is estimated by the following function E.

  $$E = \sum_{i=1}^{N_{Bk}} ImpBk_i * \frac{1}{N_{Op}} \sum_{j=1}^{N_{Op}} ImpOp_j * PerCh_{i,j}$$

  where
  - *ImpBk* is the importance of each block composing the page. The importance of a block in the page depends on its location, area size, content, etc.
  - *ImpOp* is the importance of changes operations (insertion, deletion, etc.) detected between the two versions. For instance, delete operation can be considered less important than an insert or an update.
  - *PerCh* is the percentage of changes (insert, delete, etc.) occurred on each block with respect to the total number of block's elements.
  - N_{Op}, N_{Bk} are respectively the number of change operations and the number of blocks in the page.
  - $\sum_{i=1}^{N_{Bk}} ImpBk_i = 1$

  The estimator E returns a normalized value between 0 and 1. This value assesses the importance of changes between two page versions. The larger the number of significant changes occurred inside important blocks is, the higher the estimated importance of changes. For more details about the algorithm Vi-DIFF used to detect changes between two versions of pages, please refer to [14]. The estimator of the importance of changes is detailed in [2].



Page Changes Pattern

Periods T	Workdays ω_k	Saturday ω_k	Sunday ω_k
[0:00-6:00]	0,2	0,1	0,2
[6:00-12:00]	0,4	0,4	0,35
[12:00-18:00]	0,6		
[18:00-24:00]	0,1	0,2	0,13

Fig. 1. Pattern Example

- **Page changes pattern**
  A pattern models the behavior of page's changes over periods of time, during for example a day. It is periodic and may depend on the day of the week and of the hour within a day. An example of pattern is shown in Figure 1. It defines the importance of changes over different periods of the day. Separate patterns can be defined for weekends.

4 Web Archive Model

In our web archive model, all the web pages are repeatedly captured individually. The crawler typically works at the granularity of a page and not at the granularity of a web site. It selects the most important pages to be refreshed from a large collection of URLs under a resource constraint, *e.g.* one page crawled per second. Pages that change very frequently with a significant modification, are visited more often. Our web archive A_S is defined as a set of archived sites A_{S_i}. A_{S_i} is a set of versions of pages downloaded from a site S_i. An interval of observation [o$_s$,o$_e$] is defined for the archive where o_s is the starting time and o_e is the ending time as shown in the Figure 2. The archive A_{S_i} is accessed by a user query $Q(t_q, A_{S_i})$ that browses the closest available page versions of a site S_i at a given time query t_q. Our work aims at improving the quality of versions returned to user for any time t_q.

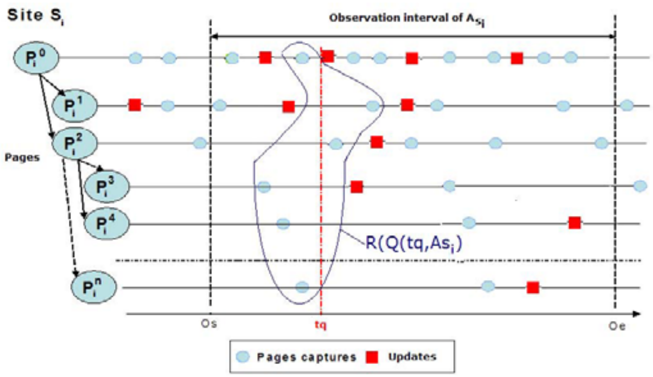


Fig. 2. Web Archive Model

4.1 Assumptions

In the following, we assume that the pages to be crawled change over time independently from each other. Patterns are considered known for any page and have already been discovered by using the approach proposed in [3]. We assume that the web crawler has limited resources for capturing new versions of pages. We model the resource constraint by assuming that the crawler can download a total of M pages in each period T. We assume that all snapshots (or states) of web sites to be crawled are coherent at each instant. This means that each site, at any time point, do not present conflicting information such as broken links, error of posting pictures, etc. This type of incoherence is out of control and is ignored in this work.

4.2 Notation and Definitions

We assume that S={S$_1$,S$_2$,...,S$_\kappa$} is the list of sites to be crawled. Each site S$_i$ consists of N_i pages {P$_i^1$,P$_i^2$,...,P$_i^{N_i}$}. Each page P$_i^j$ has a pattern Patt(P$_i^j$). In addition, we assume that the importance of the page P$_i^j$ is $\omega(P_i^j)$. The importance of a change occurred on the page P_i^j at instant t is denoted by $\omega_i^j[t]$.

Definition 4.21. *Pattern*
A pattern of a page P_i^j with an interval length l is a nonempty sequence Patt(P_i^j) = {(ω_1,T_1); ...; (ω_k,T_k);...; (ω_{N_T},T_{N_T})}, where N_T is the total number of periods in the pattern and ω_k is the average of the importance of changes estimated in the period T_k. The sum of the time periods, $\sum_{k=1}^{N_T} T_k$, is equal to l.

We note $P_i^j[t]$ the version of the page P_i^j captured at time t in the archive A_{S_i}. We note $\tilde{P}_i^j[t]$ the version of the page P_i^j created (by a change) on the real web site S_i at time t. As shown in the Figure 3, the page P_i^j has one capture at time t_1 that corresponds to the archived version $P_i^j[t_1]$. The two versions $\tilde{P}_i^j[\tau_1]$ and $\tilde{P}_i^j[\tau_2]$ created on the web site S_i correspond to the two changes occurred on the page P_i^j at time τ_1, τ_2.

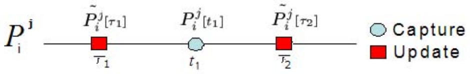


Fig. 3. Example of page versions

Definition 4.22. *Archive*
Let κ be the total number of sites, the archive A_S is the set of archived sites A_{S_i}, $S_i \in S$.

$$A_S = \bigcup_{i=1}^{\kappa} A_{S_i}$$

The archive A_{S_i} of a site S_i is defined by the set of page versions $P_i^j[t]$ captured from the site S_i during the interval [o_s, o_e].

$$A_{S_i} = \{P_i^j[t], 1 \leq j \leq N_i | P_i^j \in S_i \ \wedge t \in [o_s, o_e]\}$$

Definition 4.23. *User Query*
The user query $Q(t_q, A_{S_i})$ asks for the closest snapshot of the site S_i to the query time t_q.

Definition 4.24. *Query Result*

The result $R(Q(t_q, A_{S_i}))$ of the user query $Q(t_q, A_{S_i})$ is the set of the N_i versions $P_i^j[t]$ (one for each page of S_i) which are the closest to the time t_q as shown in Figure 2.

$$R(Q(t_q, A_{S_i})) = \{P_i^j[t] \in A_{S_i} | \neg \exists P_i^j[t'] \in A_{S_i} : |t^{'} - t_q| < |t - t_q|\}; j = \{1, .., N_i\}$$

Definition 4.25. *Version Importance*

Let $P_i^j[t]$ be the version of the page P_i^j that has been captured at time t after the change occurred at time t'. The importance $\omega(P_i^j[t])$ of the version $P_i^j[t]$ is the multiplication of the importance of its corresponding change $\omega_i^j[t']$ by the importance of the page $\omega(P_i^j)$.

$$\omega(P_i^j[t]) = \omega_i^j[t'] * \omega(P_i^j)$$

where t' is the time of the last change of $P_i^j[t]$ preceding t.

5 Quality Measures

We define, here, two quality measures completeness and coherence that take into account the importance of versions.

5.1 Completeness

The completeness of archives measures the proportion of the importance of changes that have been captured with respect to the total amount of the importance of changes that occurred on web sites.

Definition 5.11. *Complete Archive*

An archive is complete, if it contains all the versions of pages $\tilde{P}_i^j[t]$ that appeared on all sites composing the archive.

$$\forall \tilde{P}_i^j[t], \exists P_i^j[t'] \in A_{S_i}, t' \geq t : P_i^j[t'] = \tilde{P}_i^j[t]$$

Definition 5.12. *Archived Page Completeness*

The completeness of an archived page P_i^j is the sum of the weights of versions that have been captured, divided by the total weight of versions (created by changes) that appear on the real web site . Let m be respectively the number of versions $P_i^j[t_k]$ captured at time t_k and p be the number of versions $\tilde{P}_i^j[\tau_k]$ created on the site at time τ_k, the completeness of archived page P_i^j is

$$Completeness(P_i^j) = \frac{\sum_{k=1}^{m} \omega(P_i^j[t_k])}{\sum_{k=1}^{p} \omega(\tilde{P}_i^j[\tau_k])}$$

where
- The weight of the version $\omega(P_i^j[t_k])$ *is equal to the last change importance* $\omega_i^j[t']$
- The weight of the version $\omega(\tilde{P}_i^j[\tau_k])$ *is equal to the last change importance* $\omega_i^j[\tau']$
- $\omega_i^j[t']$ *and* $\omega_i^j[\tau']$ *denote the importance of the changes that occurred respectively at t' and* τ' *just before the capture of the versions* $P_i^j[t_k]$ *and* $\tilde{P}_i^j[\tau_k]$*.*

Definition 5.13. *Archived Site Completeness*
The completeness of archived site A_{S_i} *is the sum of the completeness of the* N_i *pages of* S_i *(weighted by their importance) divided by the overall pages importance.*

$$Completeness(A_{S_i}) = \frac{\sum_{j=1}^{N_i} Completeness(P_i^j) * \omega(P_i^j)}{\sum_{j=1}^{N_i} \omega(P_i^j)}$$

Definition 5.14. *Archive Completeness*
The overall completeness of the archive A_S *is the average completeness of all archived sites.*

$$Completeness(A_S) = \frac{\sum_{i=1}^{\kappa} Completeness(A_{S_i})}{\kappa}$$

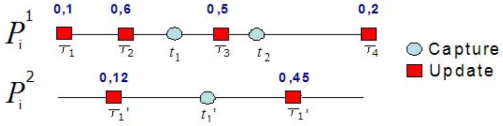


Fig. 4. Site Completeness Example

Example 5.11. *Archived Site Completeness.*
We consider a site S_i *consisting of two pages* P_i^1 *and* P_i^2 *as shown in Figure 4. We assume that* $P_i^1[t_1]$ *and* $P_i^1[t_2]$ *are the two versions of the page* P_i^1 *captured respectively at time* t_1 *and* t_2*. We assume that the importance of the four changes that have occurred on the page* P_i^1 *at time* τ_1*,* τ_2*,* τ_3 *and* τ_4 *are respectively 0.1, 0.6, 0.5, 0.2. For the page* P_i^2*, there is one capture* $P_i^2[t_1^{'}]$ *at* $t_1^{'}$*. The importance of the two changes occurred on the page* P_i^2 *at* $\tau_1^{'}$ *and* $\tau_2^{'}$ *are respectively 0.12 and 0.45.*

Note that in this example (and also in example 5.21), we assume that the importance of each page is equal to 1 $(\omega(P_i^j) = 1, \forall i, j)$.
The completeness of P_i^1 and P_i^2 is

$$Completeness(P_i^1) = \frac{\omega(P_i^{1\,[t_1]})+\omega(P_i^{1\,[t_2]})}{\sum_{k=1}^{4} \omega(P_i^{1\,[\tau_k]})} = \frac{0.6+0.5}{0.1+0.6+0.5+0.2} = 0.78$$

$$Completeness(P_i^2) = \frac{\omega(P_i^{2\,[t_1^{'}]})}{\omega(P_i^{2\,[\tau_1^{'}]})+\omega(P_i^{2\,[\tau_2^{'}]})} = \frac{0.12}{0.12+0.45} = 0.21$$

The overall completeness of archived site A_{S_i} is

$$Completeness(A_{S_i}) = \frac{Completeness(P_i^1)*\omega(P_i^1)+Completeness(P_i^2)*\omega(P_i^2)}{\omega(P_i^1)+\omega(P_i^2)}$$

$$Completeness(A_{S_i}) = \frac{0.78*1+0.21*1}{2} = 0.49$$

5.2 Coherence

A collection of archived pages versions is considered *coherent*, if it reflects the state (or the snapshot) of the web site at, at least, one point in time. Our definition of coherence is inspired by the approach of Spaniol and al. In [16], they introduce two measures to quantify the coherence of a site crawl. Their measures count the expected number of occurring incoherences during a complete crawl of a site. They are based on either (i) the last modified stamp or (ii) on a virtual time stamp obtained by revisiting each page. We do not use these measures because the last modified stamp is not always trustful in real life crawls. The virtual time stamp assumes that, during an on-line crawl, each page must be revisited twice in a short time. As we assume that the crawler has limited resources, we do not use virtual time stamps. We propose a new measure, inspired by Spaniol's definition [16], that considers the importance of changes to quantify the coherence of the query result.

Definition 5.21. *Coherent Versions*
The N_i versions of $R(Q(t_q, A_{S_i}))$ are coherent, if there is a time point (or an interval) so that it exists a non-empty intersection among the invariance interval $[\mu_j, \mu_{j^*}]$ of all versions.

$$\forall P_i^j[t] \in R(Q(t_q, A_{S_i})), \exists t_{coherence} : t_{coherence} \in \bigcap_{j=1}^{N_i} [\mu_j, \mu_{j^*}] \neq \varnothing \qquad (1)$$

where μ_j and μ_j^* are respectively the previous and the next changes following the capture of the version $P_i^j[t]$.

As shown in Figure 5 at the left, the three versions $P_i^1[t_1]$, $P_i^2[t_2]$ and $P_i^3[t_3]$ of $R(Q(t_q, A_{S_i}))$ are coherent because there is an interval $t_{coherence}$ that satisfies the coherence constraint (1). However the three page versions at the right are not coherent because there is no point in time satisfying the coherence constraint (1).

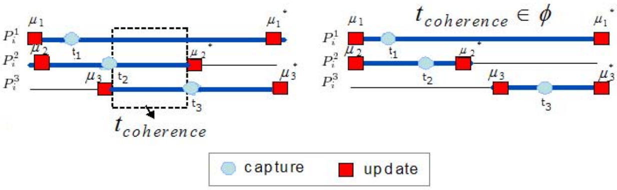


Fig. 5. Coherence Example [16]

Definition 5.22. *Query Result Coherence*
The coherence of the query result $R(Q(t_q, A_{S_i}))$ is the weight of the largest number of coherent versions divided by the total weight of the N_i versions of $R(Q(t_q, A_{S_i}))$. We assume that $\{P_i^1[t_1]$,..., $P_i^\rho[t_\rho]\} \in R(Q(t_q, A_{S_i}))$ are the ρ coherent versions, i.e satisfying the constraint (1). We assume that ρ is the

largest number of coherent versions composing $R(Q(t_q, A_{S_i}))$.
The coherence of $R(Q(t_q, A_{S_i}))$ is

$$Coherence(R(Q(t_q, A_{S_i})) = \frac{\sum_{k=1}^{\rho} \omega(P_i^k[t_k])}{\sum_{k=1}^{N_i} \omega(P_i^k[t_k])}$$

where $\omega(P_i^k[t_k])$ is the importance of the version $P_i^k[t_k]$.

Definition 5.23. *Site Archive Coherence*
The overall coherence of archived site A_{S_i} can be estimated through the average coherence of $R(Q(t_q, A_{S_i}))$ obtained for different time query t_q.

$$Coherence(A_{S_i}) = \frac{\sum_{1}^{n_Q} Coherence(R(Q(t_q, A_{S_i})))}{n_Q}$$

where n_Q is the number of queries that have accessed the archive A_{S_i} in the observation interval $[o_s, o_e]$.

Definition 5.24. *Archive Coherence*
The overall coherence of the archive A_S is the average coherence of the κ archived sites.

$$Coherence(A_S) = \frac{\sum_{i=1}^{\kappa} Coherence(A_{S_i})}{\kappa}$$

Example 5.21. *Query Result Coherence*

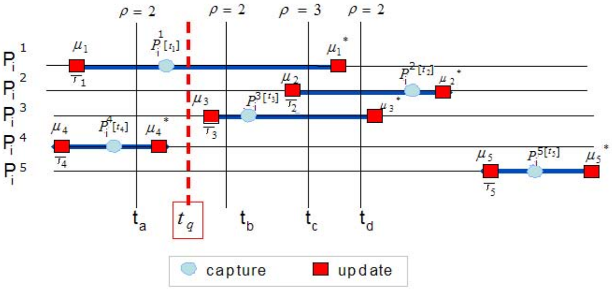


Fig. 6. $R(Q(t_q, A_S))$ Example

We assume that $P_i^1[t_1]$, $P_i^2[t_2]$, $P_i^3[t_3]$, $P_i^4[t_4]$, $P_i^5[t_5]$ are the five versions of $R(Q(t_q, A_{S_i}))$ which are the closest to the time query t_q. We assume that $\omega_i^1[\tau_1] = 0.2$, $\omega_i^2[\tau_2] = 0.54$, $\omega_i^3[\tau_3] = 0.34$, $\omega_i^4[\tau_4] = 0.22$ and $\omega_i^5[\tau_5] = 0.6$ are the importance of changes $\mu_1, ..., \mu_5$ occurred respectively on the pages P_i^1, ...,P_i^5 at instants $\tau_1, ..., \tau_5$.

The value ρ of the largest number of coherent versions is equal to 3 because there are three coherent versions $P_i^1[t_1]$, $P_i^2[t_2]$ and $P_i^3[t_3]$ satisfying the constraints (1) : it exists a time $t_c \in [\mu_1, \mu_1^*] \bigcap [\mu_2, \mu_2^*] \bigcap [\mu_3, \mu_3^*]$.
Then, the coherence of $R(Q(t_q, A_{S_i}))$ is

$$Coherence(R(Q(t_q, A_{S_i}))) = \frac{\omega(P_i^1[t_1]) + \omega(P_i^2[t_2]) + \omega(P_i^3[t_3])}{\sum_{k=1}^{5} \omega(P_i^k[t_k])}$$

$$= \frac{0.2 * 1 + 0.54 * 1 + 0.34 * 1}{0,2 * 1 + 0.54 * 1 + 0.34 * 1 + 0.22 * 1 + 0.6 * 1} = 0.56$$

6 Pattern-Based Web Crawling

To improve the quality of archives, we propose a crawling strategy directly driven by the patterns defined in Section 4.2. Our goal is to schedule page crawls in such a way that it improves completeness and coherence of the archive. The crawler can download a total of M pages at each period T_k. To simplify notations, we assume, in the remainder of the paper, that $P_1, P_2, ..., P_n$ is the list of all pages to be crawled. By using patterns, pages are scheduled based on their urgency (or priority). Each page is assigned an urgency value *U(P_i,t)* proportional to both the expected changes importance ω_k (defined by pattern at period T_k) and to the importance of the page $\omega(P_i)$. Also, the urgency of pages changes with the time. It depends on the time of the last refresh and on the current time. The urgency *U(P_i,t)* of the page P_i at time t is
$U(P_i, t) = \omega(P_i) * \omega_k * (t - t_{lastRefresh})$
where

- Patt(P$_i$) = $\{(\omega_1,T_1)$; ...; (ω_k,T_k);...; $(\omega_{N_T},T_{N_T})\}$,
- t is the current time (t $\in T_k$),

Algorithm 1. Pattern-based Crawler

```
Input:
P_1, P_2,..., P_n - list of pages
Patt(P_1), Patt(P_2),..., Patt(P_n) - patterns of pages
Begin
for each period $T_k$ do
  crawlListPages⟵newList()
  for each page P_i, i=1...,n do
    compute $U(P_i, t) = \omega(P_i) * \omega_k * (t_i - t_{ilastRefresh})$
    crawlListPages.add(P_i,$U(P_i, t)$) /* in descending order of urgency */
  end for
  for i=1...,M do
     P_i⟵crawlListPages.selectPage(i)
     currentVersion⟵downloadPage(P_i)
     lastVersion⟵getLastVersion(P_i)
     delta⟵detectChanges(currentVersion,lastVersion)
     $\omega$⟵EstimateChangesImportance(delta)
     Update(Patt(page),$\omega$,$T_k$)
     $t_{ilastRefresh} \longleftarrow t_i$
  end for
end for
End
```

- ω_k is the average of change importance defined by Patt(P$_i$) in period T_k,
- $\omega(P_i)$ is the importance of the page,
- $t_{lastRefresh}$ is the last time of refreshing the page P_i.

At each period T_k, only the M-top pages with the highest current priority are captured. The M selected pages are downloaded in descending order of their urgency $U(P_i, t)$. Afterwards, each captured page version is compared with its predecessor to detect changes based on the Vi-DIFF algorithm (*cf.* Section 3). Then, the importance of changes can be estimated by the function E (*cf.* Section 3) and exploited to update patterns. Patterns need to be updated periodically to always reflect the current changes of web pages. Thus, the average change importance ω_k defined by patterns in period T_k is periodically updated during an on-line crawl. Also, the importance of page (*e.g.* PageRank) is regularly reevaluated over time to reflect the real web. The pseudo code of the pattern-based strategy is depicted by Algorithm 1.

7 Experimental Evaluation

In this section, we evaluate the effectiveness of our crawling approach by comparing it with existing strategies. In particular, we compare the total completeness and coherence (*cf.* Section 5) obtained by each policy. As it is impossible to obtain exactly all the versions that appear on real web sites, we have simulated the change importance of web pages based on real patterns discovered from "France Télévisions" channels pages [3]. Experiments written in Java were conducted on PC running Linux over a 3.20 GHz Intel Pentium 4 processor with 1.0 GB of RAM. At the begin of each experiment, each page is described by a real pattern. The updates rate and the changes importance of each page is generated according to defined patterns. In addition, the following parameters are set: the number of pages per site (one thousand pages), the duration of simulation, the number of periods in patterns (24 periods), the number of allocated resources (*i.e.* the maximum number of pages that can be captured per each time period). Equal resources are assigned to different crawler strategies to evaluate them under the same constraints.

We start by describing related strategies considered in this work: ***Relevance*** [9] downloads the most important pages (*i.e* based on PageRank) first, in a fixed order. ***Frequency*** [7] selects pages to be archived according to their frequency of changes estimated by the Poisson model [8]. Hot pages that change too often are penalized to maximize the freshness of pages. ***Coherence*** [16] works at the granularity of a site and downloads firstly the less "risky" pages (*i.e.* with lowest probability to cause incoherences). Then, it continues by capturing the remaining pages that had been skipped. ***SHARC*** [10] repeatedly downloads the entire sites and ensures that the most changing page are downloaded as close as possible to the middle of the capture interval. ***Importance*** is our first strategy which selects pages based on their urgency (*cf.* Section 6). The parameter of changes importance ω_k is a fixed weight (average) and does not depend on time periods

T_k. This strategy consider the importance of changes without using patterns. ***Pattern*** is our second strategy which depends only on patterns without considering the importance of changes. It downloads pages according to their urgency based on changes rate instead on changes importance. ***Importance-Pattern*** is our third strategy which downloads pages based on their urgency (*cf.* Algorithm 1). It combines the two concepts importance of changes and patterns.

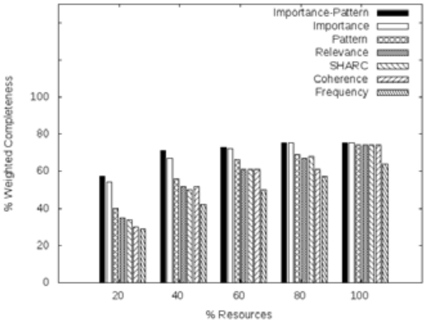


Fig. 7. Weighted Completeness

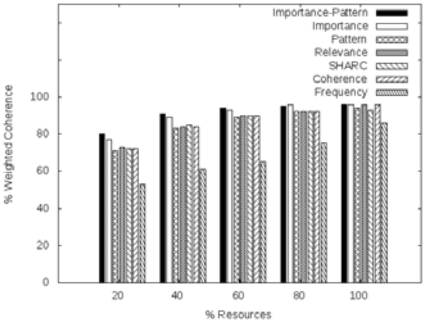


Fig. 8. Weighted Coherence

We evaluated the completeness (*cf.* Section 5.1) obtained by the different crawl strategies as shown in the Figure 7. The horizontal axis shows the percentage of allocated resources M=[20%-100%]. The vertical axis shows the weighted completeness that have been captured by each policy. As we can see, the completeness increases with the number of allocated resources. Obviously, when more pages are captured at each period, better completeness is achieved. We notice also that it is impossible to achieve 100% of completeness even if 100% of resources are allocated. There are always some missed versions. From the figure, it is clear that our strategy *Importance-Patterns* performs better than its competitors *Relevance*, *SHARC*, *Coherence* and *Frequency*. It improves the completeness of the archive by around 20% in case of limited resources. Also, it performs 5% better than *Importance* strategy which does not use patterns. This gain is rather low in average, but we note that it can reach more than 10 % in case the patterns of web pages are significantly different one from the others.

We evaluated also the coherence of the archive based on the measure defined in Section 5.2. Figure 8 shows the percentage of coherence weight achieved by different crawl strategies with respect to the number of allocated resources. As we can see, our *Importance-Patterns* strategy achieves the highest coherence weight. It performs around 10% better than its competitors *SHARC*, *Coherence* and *Relevance*. *Frequency* strategy achieves the lowest coherence weight. To sum up, these experiments demonstrate that *Importance-Patterns* improves both the completeness and the coherence of archives by respectively 20 % and 10 %.

8 Conclusion and Future Work

Preserving the quality of web archives is a crucial issue addressed by archivists nowadays. Here, we point out the issue of efficiently crawling web pages in order to improve the quality of archives. We defined two metrics to measure the quality of the archive. Completeness measures the ability of the archive to contain the largest amount of useful information. Coherence measures how much the archive reflect the snapshot of web sites at different points in time. As far as we know, this work is the first to formalize and to address both issues (coherence and completeness) at the same time. Our challenge is to adjust the crawl strategy to make the archive as complete and as coherent as possible. We propose a pattern-based strategy which use the importance of changes to improve the quality of the archive. To the best of our knowledge, the concepts of the importance of changes and patterns have never been used to improve the quality of archives. Most related strategies download with priority the most frequently changing pages (and/or the most important ones based on PageRank) but do not consider the importance of changes occurred between page versions. Conducted experiments based on real patterns confirm that our pattern-based strategy outperforms its competitors. Results show that it is able to improve the completeness of the archive by around 20% and the coherence by around 10 % in case of limited resources. This improvement of the archive quality can be furthermore better, if patterns of web pages are significantly different one from the others.

We are currently pursing our study to run our pattern-based strategy over a large number of web pages collected by the National Audiovisual Institute (INA). We are also studying how patterns can be exploited to decide when page versions should be indexed or stored. Hence, archive systems will avoid wasting time and space for indexing/storing unimportant pages versions. Further study must be done to learn how we can create a collection of common patterns for pages with similar behavior of changes. An other on-going work is to find an efficient method to maintain patterns up-to-date during an on-line crawl.

References

1. Adar, E., Teevan, J., Dumais, S.T., Elsas, J.L.: The web changes everything: understanding the dynamics of web content. In: Proceedings of the Second ACM International Conference on Web Search and Data Mining, Barcelona, Spain (2009)
2. Ben Saad, M., Gançarski, S.: Using visual pages analysis for optimizing web archiving. In: EDBT/ICDT PhD Workshops, Lausanne, Switzerland (2010)
3. Ben Saad, M., Gançarski, S.: Archiving the Web using Page Changes Pattern: A Case Study. In: ACM/IEEE Joint Conference on Digital Libraries (JCDL 2011), Ottawa, Canada (2011)
4. Brewington, B.E., Cybenko, G.: Keeping up with the changing web. Computer 33(5) (2000)
5. Castillo, C., Marin, M., Rodriguez, A., Baeza-Yates, R.: Scheduling algorithms for web crawling. In: LA-WEBMEDIA 2004: Proceedings of the WebMedia (2004)

6. Cho, J., Garcia-Molina, H.: The Evolution of the Web and Implications for an Incremental Crawler. In: VLDB 2000: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 200–209. San Francisco, CA, USA (2000)
7. Cho, J., Garcia-Molina, H.: Effective page refresh policies for web crawlers. ACM Trans. Database Syst. 28(4), 390–426 (2003)
8. Cho, J., Garcia-molina, H.: Estimating frequency of change. ACM Transactions on Internet Technology 3, 256–290 (2003)
9. Cho, J., Garcia-molina, H., Page, L.: Efficient crawling through url ordering. In: Computer Networks and ISDN Systems, pp. 161–172 (1998)
10. Denev, D., Mazeika, A., Spaniol, M., Weikum, G.: Sharc: framework for quality-conscious web archiving. Proc. VLDB Endow. 2(1), 586–597 (2009)
11. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. In: Data Mining and Knowledge Discovery, vol. 15 (2007)
12. Masanès, J.: Web Archiving. Springer, New York (2006)
13. Olston, C., Pandey, S.: Recrawl scheduling based on information longevity. In: Proceeding of the 17th International Conference on World Wide Web (2008)
14. Pehlivan, Z., Ben-Saad, M., Gançarski, S.: Vi-DIFF: Understanding web pages changes. In: Bringas, P.G., Hameurlain, A., Quirchmayr, G. (eds.) DEXA 2010. LNCS, vol. 6261, pp. 1–15. Springer, Heidelberg (2010)
15. Sia, K.C., Cho, J., Cho, H.-K.: Efficient monitoring algorithm for fast news alerts. IEEE Transactions on Knowledge and Data Engineering 19, 950–961 (2007)
16. Spaniol, M., Denev, D., Mazeika, A., Weikum, G., Senellart, P.: Data quality in web archiving. In: WICOW 2009: Proceedings of the 3rd Workshop on Information Credibility on the Web, pp. 19–26 (2009)
17. Spaniol, M., Mazeika, A., Denev, D., Weikum, G.: "catch me if you can": Visual analysis of coherence defects in web archiving. In: 9th International Web Archiving Workshop (IWAW 2009), Corfu, Greece, pp. 27–37 (2009)