Coherence-oriented Crawling and Navigation using Patterns for Web Archives *

Myriam Ben Saad, Zeynep Pehlivan, and Stéphane Gançarski

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

Abstract. We point out, in this paper, the issue of improving the coherence of web archives under limited resources (*e.g.* bandwidth, storage space, etc.). Coherence measures how much a collection of archived pages versions reflects the real state (or the snapshot) of a set of related web pages at different points in time. An ideal approach to preserve the coherence of archives is to prevent pages content from changing during the crawl of a complete collection. However, this is practically infeasible because web sites are autonomous and dynamic. We propose two solutions: *a priori* and *a posteriori*. As *a priori* solution, our idea is to crawl sites during the *off-peak* hours (*i.e.* the periods of time where very little changes is expected on the pages) based on patterns. A pattern models the behavior of the importance of pages changes during a period of time. As an *a posteriori* solution, based on the same patterns, we introduce a novel navigation approach that enables users to browse the most coherent page versions at a given query time.

Keywords: Web Archiving, Data Quality, Pattern, Navigation

1 Motivation

The major challenge of web archiving institutes (Internet Archive, etc.) is to collect, preserve and enable future generations to browse off-line a rich part of the Web even after it is no more reachable on-line. However, maintaining a good quality of archives is not an easy task because the web is evolving over time and allocated resources are usually limited (*e.g.* bandwidth, storage space, etc.). In this paper, we focus on the coherence that measures how much a collection of archived pages versions reflects the real web at different points in time. When users navigate through the archive, they may want to browse a collection of related pages instead of individual pages. This collection is a set of linked pages which may or not share the same context, topic, domain name, etc. It can be a web site generally including a home page and located on the same server. But it can be also a set of interconnected web pages belonging to different sites. Coherence ensures that if users reach a page version, they can also reach to the versions

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

of other pages of the same collection, corresponding to the same point in time. In fact, during the navigation in the archive, users may browse a page version which refers to another page, but the two page versions have never appeared at the same time on the real web. This may lead to conflicts or inconsistencies between page versions, and such pages versions are considered temporally incoherent. Figure 1 depicts an example of incoherence while brwosing at time t_q.We consider two pages P_1 and P_2 of a site, updated respectively at time t_2 and t_5. A and A' are the two versions of the page P_1 captured at time t_1 and t_3. B is the only version of the page P_2 captured at t_4. If the archive is queried at time t_q, we can obtain as result the two versions A' and B because they are the "closest" from t_q. These two versions are not coherent because they have never appeared at the same time on the site (caused by pages update). However, the two versions A and B are coherent because they appear "as of" time point t_1.

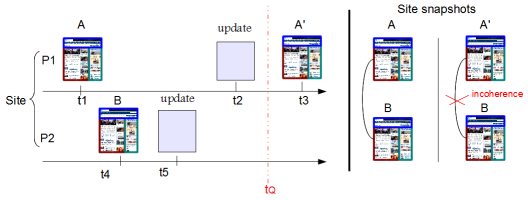

Fig. 1. Archive Coherence

The problem of incoherence usually happens when pages change their content during the crawl of an entire collection. This problem can be handled by two solutions: *a priori* and *a posteriori*. The *a priori* solution aims to minimize pages incoherence at the crawling time by adjusting crawlers strategy. The *a posteriori* solution operates at the browsing time by enabling users to navigate through the most coherent pages versions. The *a priori* solution adjusts crawlers strategy to maximize the coherence of collected pages versions independently of other quality measures (*e.g.* completeness, freshness, etc.). As it is impossible to obtain 100% of coherence in the archive due to the limited resources, an *a posteriori* solution is also needed. Thus, our challenge is to optimize browsing to enable users to navigate through the most coherent page versions at a given query time. In [2], we have discovered periodic patterns from TV channels pages which describe the behavior of (regular) changes over time. By exploiting these patterns, we propose, in this paper, novel coherence-oriented approaches of crawling and browsing to improve archives quality and users navigation.

This paper is structured as follows. In Section 2, related works are discussed. Section 3 defines a coherence measure. Section 4 describes web archiving model based on pattern. Section 5 proposes a crawling strategy to maximize archives coherence. Section 6 introduces a coherence-oriented approach to improve archive navigation. Section 7 presents preliminary results. Section 8 concludes.

2 Related Works

In recent years, there has been an increasing interest in improving coherence of web archives. In [13], authors propose a crawling strategy to improve coherence

of crawled sites. However, they do not mention in which order sites should be visited. In [14], they present visualization strategies to help archivists to understand the nature of coherence defects in the archive. In another study, they define two quality measures (blur and sharp) and propose a framework, coined SHARC, to optimize pages captures. The two policies [13, 9] are based on multiple revisits of web pages. However, in our work, we assume that web crawlers have limited resources which prevent from revisiting pages too often. Other studies are also closely related to our work in the sense that they aim at optimizing crawlers. To guess at which frequency each page should be visited, crawl policies are based on three major factors: (i) the relevance /importance of pages (e.g Page rank) [7], (ii) information longevity [11] and (iii) frequency of changes [5, 9]. A factor that has been ignored so far is the importance of changes between pages versions. Moreover, the frequency of changes used by most policies is estimated based on homogenous poisson process which is not valid when pages are updated frequently as demonstrated in [3]. Our research is applied on the archive of French National Institute (INA) which preserves national radio and TV channels pages. These pages are updated several times a day and, hence, the poisson model can not be used as explained above. In [2], we discovered periodic patterns from TV channels pages by using statistical analysis technique. Based on patterns, we propose, in this paper, a crawl policy to improve the coherence of archives.

This paper also presents a new navigation method that takes into account the temporal coherence between the source page and the destination page. Although there are several browsers proposed to navigate over historical web data [10, 15], they are only interested in navigation between versions of the same pages by showing the changes over versions. As far as we know,no approach proposes to improve the navigation in web archives by taking into account temporal coherence. The reason, as explained in [4], can be that temporal coherence only impacts the very regular users who spend lots of time navigating in the web archives. Even though, today the archive initiatives do not have many users, we believe that, popular web archives (e.g Internet Archive, Google News Archive) will get the attention of more and more regular users over web archives.

3 Coherence Measure

We define in this section a quality measure inspired by [13] which assesses the coherence of archives. The following notations are used in the paper.

- S_i is a collection of linked web pages P_i^j.
- A_{S_i} is a (historical) archive of S_i.
- $P_i^j[t]$ is a version of a page P_i^j ($P_i^j \in$ collection S_i) captured at time t.
- $\mathrm{Q}(t_q, A_{S_i})$ is a query which asks for the closest versions (or snapshot) of A_{S_i} to the time t_q.
- $R(Q(t_q, A_{S_i}))$ is a set of versioned pages obtained as a result of querying A_{S_i} at time t_q. A' and B in Figure 1 both belong to $\mathrm{R}(\mathrm{Q}(t_q, A_S))$.
- $\omega(P_i^j[t])$ is the importance of the version $P_i^j[t]$. It depends on (i) the weight of the page P_i^j (*e.g.* PageRank) and on (ii) the importance of changes between

$P_i^j[t]$ and its last archived version. The importance of changes between two pages versions can be evaluated based the estimator proposed in [1].

Definition 1 *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) called $t_{coherence}$, 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 time points of the previous and the next changes following the capture of the version $P_i^j[t]$.

As shown in Figure 2, the three versions $P_i^1[t_1]$, $P_i^2[t_2]$ and $P_i^3[t_3]$ 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).

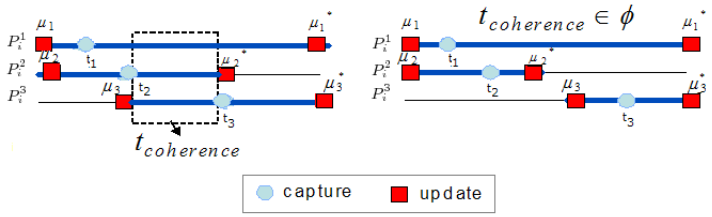

Fig. 2. Coherence Example [13]

Definition 2 *Query-Result Coherence*

The coherence of the query result $R(Q(t_q, A_{S_i})$, also called weighted coherence, 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). 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 $P_i^k[t_k]$.

Instead of evaluating the coherence of all the versions composing the query result $R(Q(t_q, A_{S_i}))$, we can restrict the coherence measure to only the η-top pages of A_{S_i} which are the most relevant ones. Such measure is useful to preserve particularly the coherence of the most browsed (important) pages like home pages and their related pages. Coherence of rarely browsed pages can be considered less important.

4 Pattern Model

Our work aims at improving the coherence of archived web collections by using patterns. We describe here the pattern model.

4.1 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. Pages with similar changes behavior can be grouped to share a common pattern.

Definition 3 *Pattern*

A pattern of a page P_i^j *with an interval length* l *is a nonempty sequence* $Patt(P_i^j) = \{(\omega_1,T_1); ...; (\omega_k,T_k);...; (\omega_{N_T},T_{N_T})\}$*, where* N_T *is the total number of periods in the pattern and* ω_k *is the estimated importance of changes in the period* T_k*.*

4.2 Pattern-based Archiving

As shown in Figure 3, patterns are discovered from archived page versions by using an analyzer. The first step of the analyzer consists on segmenting each captured pages into blocks that describe the hierarchical structure of the page. Then, successive versions of a same page are compared to detect *structural*[1] and *content*[2] changes by using Vi-DIFF algorithm [12]. Afterwards, the importance of changes between two successive versions is evaluated based on the estimator proposed in [1]. This estimator returns a normalized value between 0 and 1. An importance value near one (respectively near 0) denotes that changes between versions are very important (respectively irrelevant *e.g.* advertisements or decoration). After that, a periodic pattern which models changes importance behavior is discovered for each page based on statistical analysis. In [2], we have presented, through a case study, steps and algorithms used to discover patterns from French TV channels pages. Discovered patterns are periodically updated to always reflect the current behavior. They can be used to improve the coherence of archives. Also, they can be exploited by the browser to enable users to navigate through the most coherent page versions as shown in Figure 3.

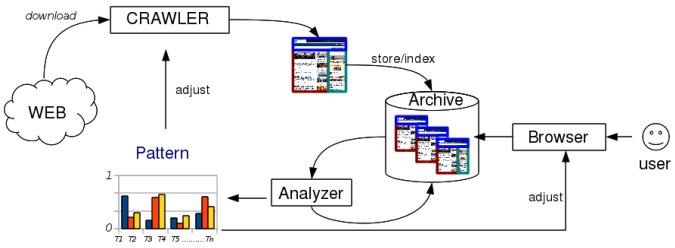

Fig. 3. Pattern-based Archiving

5 Coherence-oriented Crawling

An ideal approach to preserve the coherence of archives is to prevent pages content from changing during the crawl of a complete collection. As this is practically impossible, we have the idea to crawl each colection during the periods of

[1] The changes that affect the structure of blocks composing the page
[2] The changes that modify links, images and texts inside blocks of the pages

time where very little (or useless) changes are expected to occur on pages. Such periods are named *off-peak periods*. Based on discovered patterns, these periods can be predicted for each page and grouped to share a common off-peak period for the collection as shown in Figure 4. To improve the coherence, it is better to start by crawling S_2 before S_1 in order to coincide with their *off-peak* periods (Figure 4).

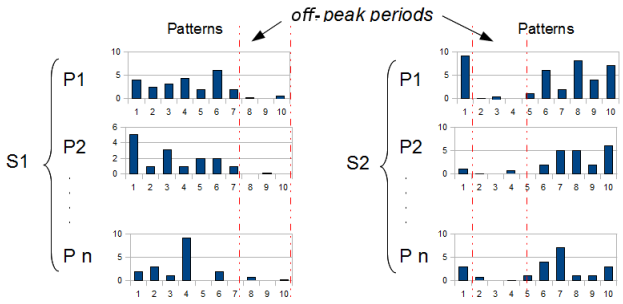

Fig. 4. Crawling collections at *off-peak periods*

5.1 Crawling Strategy

Given a limited amount of resources, our challenge is to schedule collections according to their off-peak periods in a such way that it improves the coherence of the archive. We define an urgency function that computes the priority of crawling a collection S_i at time t. The urgency $U(S_i,t,\eta)$ of crawling the collection S_i at time t is

$$U(S_i, t, \eta) = [1 - \varphi(S_i, T_k, \eta)] * (t - t_{lastRefresh})$$

- t is the current time (t $\in T_k$),
- $t_{lastRefresh}$ is the last time of refreshing the collection S_i.
- η is the number of pages considered to evaluate the coherence of A_{S_i}
- $\varphi(S_i, T_k, \eta)$ is the average of the importance of the changes predicted by patterns during the period T_k for the collection S_i.

$$\varphi(S_i, T_k, \eta) = \frac{\sum_{k=1}^{\eta} \omega_k}{\eta}$$

where ω_k is the importance of changes defined in Patt(P_i^j) ($1 \leq j \leq \eta$) at T_k.

The urgency of a collection depends on the importance of changes predicted by patterns and also on the duration between the current and the last refresh time. Less important changes occur in period T_k, higher is the priority given to crawl the collection S_i. Only the M-top collections with the highest priority are downloaded at each period T_k. The value M is fixed according to available resources (*e.g.* bandwidth, etc.). Once the M collections to be crawled are selected, the different pages are downloaded in descending order of their importance changes predicted by their patterns in period T_k. It is better to start by crawling pages with the highest changes importance because the risk of obtaining an incoherence heavily depends on the time of downloading each page.

Capturing static pages at the end of crawl period does not affect the coherence of archived collection. A pseudo code of the implementation of this strategy is depicted by Algorithm 1.

Algorithm 1 Coherence-oriented Crawling

Input:
S_1, S_2,..,S_i,..., S_N - list of collections
Patt(P_i^1),Patt(P_i^2),..., Patt(P_i^j),..., Patt(P_i^n) - list of Page patterns
Begin
1. **for** each collection S_i, i=1...,N in period T_k **do**
2. compute $U(S_i, t, \eta) = [1 - \varphi(S_i, T_k, \eta)] * (t - t_{lastRefresh})$
3. collectionList.add(S_i,$U(P_i, t)$) /* in descending order of urgency */
4. **end for**
5. **for** i=1...,M **do**
6. S_i ⟵collectionList.select(i)
7. $t_{lastRefresh}$ ⟵ t
8. pageList⟵getPagesofCollection(S_i)
9. reorder(pageList, w_k) /* in descending order of changes importance */
10. **for** each page P_i^j in pageList **do**
11. download page P_i^j
12. **end for**
13. **end for**
End

6 Coherence-oriented Navigation

In web archives, navigation, also known as surfing, is enriched with the temporal dimension. In [10], web archive navigation is represented in two different categories: horizontal navigation and vertical navigation. Horizontal navigation lets users to browse chronologically among different versions of a page, while vertical navigation lets users to browse by following hyperlinks between pages like in the web. In this paper, we are interested in vertical navigation. Although it looks like navigation in the real web, the issues induced by web archives (temporal coherence and incompleteness) lead to broken or defected links which disable the complete navigation. As we know, it is impossible to obtain 100 % of coherence in the archive because allocated resources are usually limited and pages are too dynamic. If the system does not hold a version that was crawled exactly at the requested time, it usually returns the nearest (or recent) version. Even by finding the nearest version from the multi archives view, like in Memento framework [8], it is not sure that thuis version reflects the navigation like it was in real web. We introduce here a navigation approach that enables users to navigate through the most coherent versions.

In the remainder of the paper, the notion of collections of pages are not used anymore because while navigating in the archive, we focus on the coherence of two linked pages: (i) the source page and (ii) the destination page pointed by an hyperlink from the source page. In the following, a page is denoted by P_j and a version of the page crawled at instant t is denoted by $P_j[t]$.

6.1 Informal Overview

A simple example is given in Figure 5 to better explain our coherence-oriented navigation approach. Consider a user who starts to navigate in the archive from

the version of the page P_1 captured at t_q ($P_1[t_q]$). This user wants to follow the hyperlink to browse the page P_2. The closest version of P_2 before t_q is $P_2[t_1]$ and the closest version of P_2 after t_q is $P_2[t_2]$. They are the candidate destination versions. We assume that the patterns of P_1 and P_2 which describe the behavior of changes are known. These patterns are used to decide which version is the most coherent. As shown in Figure 5, the subpatterns, defined according to the periods $[t_1, t_q]$ (in red) and $[t_q, t_2]$ (in green), are extracted from patterns of P_1 and P_2. To find the most coherent version to $P_1[t_q]$, we estimate the importance of changes for each subpattern. Smaller the importance of changes predicted by subpatterns is, smaller the risk of incoherence. Thus, the group of subpatterns (a) and (b) is compared to other group of subpatterns (c) and (d) by using the importance of changes. The group of subpatterns which has the smallest total importance of changes is selected. This means that the navigation through the corresponding page versions in the selected group is more coherent. In the example, the group of subpatterns (a) and (b) has smaller importance of changes than the group of (c) and (d). Thus, the most coherent version $P_2[t_1]$ (corresponding to subpattern (b)) is returned to the user.

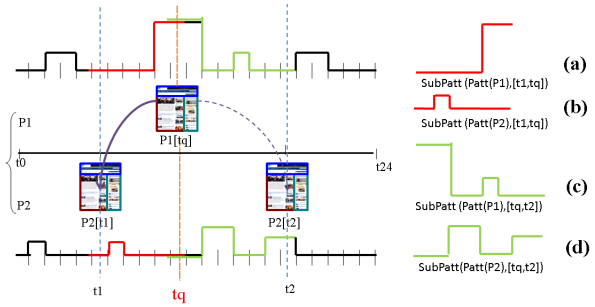

Fig. 5. Coherence-oriented Navigation

6.2 Formal Definitions

In this section, we give the formal definitions of our approach explained in the previous section.

Definition 4 *SubPattern*

Given a pattern $Patt(P_j) = \{(\omega_1,T_1); ...; (\omega_k,T_k);...; (\omega_{N_T},T_{N_T})\}$, *the subpattern* $SubPatt(Patt(P_j), [t_x, t_y])$ *is a part of the pattern valid for a given period* $[t_x, t_y]$.

$$SubPatt(Patt(P_j), [t_x, t_y]) = \{(\omega_k, T_k); (\omega_{k+1}, T_{k+1}); ...; (\omega_l, T_l)\}$$
$$where\ 1 \leq k \leq l \leq N_T \quad and\ t_x \in T_k\ and\ t_y \in T_l$$

Definition 5 *Pattern Changes Importance*

The function $\Psi(Patt(P_j))$ *estimates the total importance of changes defined in the given pattern* $Patt(P_j)$. *It is the sum of all changes importance* ω_i *of* $Patt(P_j)$.

$$\Psi(Patt(P_j)) = \sum_{i=k}^{i \leq l} \omega_i$$

Definition 6 *Navigational Incoherence*
Let $P_s[t_q]$ be the source version where the navigation starts. Let $P_d[t_x]$ be the destination version ($P_d[t_x]$) pointed by an hyperlink. The navigational incoherence (Υ) between the two versions $P_s[t_q]$ and $P_d[t_x]$ is the sum of changes importance predicted by their corresponding subpatterns during the period $[t_q, t_x]$. t_q and t_x are respectively the instants of capturing the source and the destination versions.

$$\Upsilon(P_s[t_q], P_d[t_x]) = \Psi(SubPatt(Patt(P_s), [t_q, t_x])) + \Psi(SubPatt(Patt(P_d), [t_q, t_x]))$$

where $t_q \leq t_x$

Definition 7 *Most Coherent Version*
To find out the most coherent destination version, the navigational incoherence (Υ) between the source version and the set of the candidate destination versions are compared and the destination version with the smallest Υ is returned. The reason to choose the smallest Υ is that the probability of being incoherent depends on the importance of changes of subpatterns. In other words, if there are less changes, the source and the destination version are expected to be more coherent. The most coherent version is described as follows:

$$MCoherent(P_s[t_q], \{P_d[t_x], P_d[t_y]\}) = \begin{cases} P_d[t_x] & \text{if } \Upsilon(P_s[t_q], P_d[t_x]) < \Upsilon(P_s[t_q], P_d[t_y]) \\ P_d[t_y] & \text{otherwise} \end{cases}$$

Example 1 *We take the same example of Figure 5 to explain the process. We assume that the importance of changes for the four subpatterns a, b, c, d are respectively 0.6, 0.1, 0.7, 0.6.*
The most coherent version $MCoherent(P_s[t_q], \{P_d[t_x], P_d[t_y]\})$ is $P_2[t_1]$ because $\Upsilon(P_1[t_q], P_2[t_1])$ is smaller than $\Upsilon(P_1[t_q], P_2[t_2])$ where

$$\Upsilon(P_1[t_q], P_2[t_1]) = 0.6 + 0.1 = 0.7 \text{ and } \Upsilon(P_1[t_q], P_2[t_2]) = 0.7 + 0.6 = 1.3$$

7 Experimental Evaluation

We evaluate here the effectiveness of the coherence-oriented crawling and navigation. As it is impossible to capture exactly all page changes occurred on web sites to measure the coherence, we have conducted simulations experiments based on real patterns obtained from French TV channels pages [2]. 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. Each page is described by its real pattern and the corresponding importance of changes is generated according to this pattern. In addition, the following parameters are set: the number of pages per collection, the duration of simulation, the number of periods in patterns, the number of allocated resources (*i.e.* the maximum number of sites (or pages) that can be captured per each time period).

7.1 Coherence-oriented Crawling Experiments

We have evaluated the coherence obtained by our *Pattern* strategy (*cf.* Algorithm 1) compared to the following related crawl policies: *Relevance* [7] which downloads first the most important sites and pages in a fixed order based on

PageRank, *SHARC* [9] which repeatedly selects in a fixed order the sites to be crawled then downloads the entire site by ensuring that the most changing pages are downloaded close to the middle of the capture interval, *Coherence* [13] which repeatedly downloads sites in a circular order. Within a site, it starts by crawling the pages that have the lowest probability to cause incoherence in the archive and *Frequency* [5] which selects sites in circular order and crawls pages according to their frequency of changes estimated by a Poisson model [6].

All experiments that have been conducted to evaluate those strategies are done under the same conditions (*i.e.* a maximum of M sites can be captured at each period T).

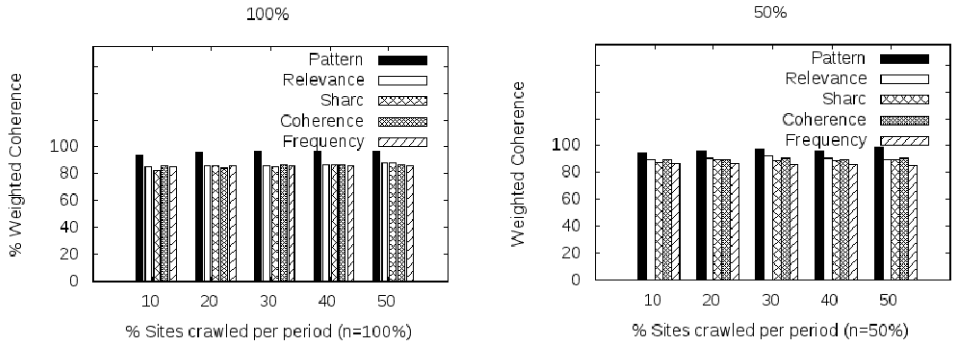

Fig. 6. Weighted Coherence

Figure 6 shows the weighted coherence (*cf.* Section 3) obtained by the different strategies with respect to the percentage of sites crawled per period M=[10%-50%]. We varied the number η of top-pages considered to evaluate the coherence of the site (η=50%,100%). As we can see, our *Pattern* strategy, which crawls collections according to their off-peak periods, outperforms its competitors *SHARC*, *Coherence*, *Relevance* and *Frequency*. It improves the coherence by around 10 % independently of the percentage of sites crawled per period. This improvement can be observed even better if the patterns of collections are significantly different from one another.

7.2 Coherence-oriented Navigation Experiments

Similarly to the crawling experiments, we implemented our navigation approach (*cf.* Section 6) over a simulated archive based on the real patterns obtained from France TV channels pages. The experiment consists in simulating the navigation from a page source P_s to different destination pages P_d by following all outgoing links from P_s. In addition, we implemented two related navigation strategies: *Nearest* and *Recent*. The *Nearest* policy enables to navigate through the closest versions to the query time t_q. The *Recent* policy enables to navigate through the closest versions before the query time t_q. The coherence of our navigation policy *Pattern* is compared to *Nearest* and *Recent* strategies based on the definition 1 of Section 3. As we use a simulator, we know which version of the destination page is the most coherent at the beginning of the experiments. For each strategy,

we count how many times the most coherent version (*i.e.* the version satisfying the coherence constraint (1)) is chosen and then this number is divided by the total number of outgoing links in the source page.

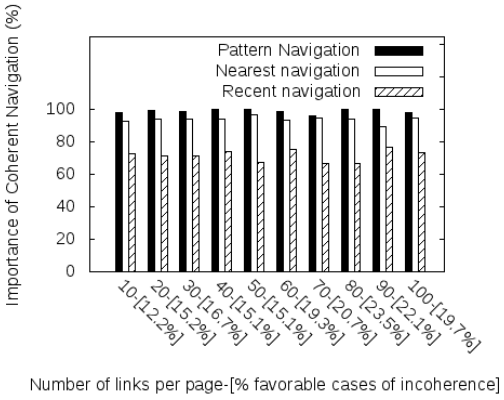

Fig. 7. Coherence-oriented Navigation

Figure 7 shows the percentage of coherent versions obtained by different strategies (*Pattern*, *Nearest*, *Recent*) with respect to the total number of outgoing links of the page source. As presented in the horizontal axis, the number of outgoing links from the page source P_s is varying from 10 to 100. We have included in brackets the percentage of cases where the nearest destination page version $P_d[t]$ to the query time t_q is incoherent with the page source version $P_s[t]$. It is important to point out that the percentage of incoherence cases presented in brackets is computed as an average obtained through several executions of simulated experiments. For example, the page source with 70 outgoing links at time t_q has about 20,7% of links where the nearest version of the destination pages are incoherent. As seen in Figure 7, our navigation policy based on patterns outperforms its competitors *Nearest* and *Recent*. It improves the coherence by around 10 % compared to *Nearest* and by around 40 % compared to *Recent*. These results are not only significant but also important since the navigation is one of the main tools used by archive users such as historians, journalists etc.

8 Conclusion and Future work

This paper addresses an important issue of improving coherence of archives under limited resources. We proposed two solutions : *a priori* and *a posteriori*. The *a priori* solution adjusts crawlers strategy to improve archive coherence by using patterns. We have demonstrated that reordering collections of web pages to crawl according to their off-peak periods can improve archives coherence by around 10 % compared to current policies in use. Moreover, as an *a posteriori* solution, we proposed a novel browsing approach using patterns that enables users to navigate through the most coherent pages versions. Results of experiments have shown that our approach can improve the coherence during the navigation by around 10 % compared to related policies *Nearest* and *Recent*. To the best of

our knowledge, this work is the first to exploit patterns to improve coherence of crawling and navigation. As a future direction, we intend to test the two proposed solutions over real data. Our challenge is to enable users to navigate through the most coherent versions at a reasonable time. Further study needs to be done to evaluate how far users can perceive coherence improvements when they navigate in the archive.

References

1. M. Ben Saad and S. Gançarski. Using visual pages analysis for optimizing web archiving. In *EDBT/ICDT PhD Workshops*, Lausanne, Switzerland, 2010.
2. M. Ben Saad and S. Gançarski. Archiving the Web using Page Changes Pattern: A Case Study. In *ACM/IEEE Joint Conference on Digital Libraries (JCDL '11)*, Ottawa, Canada, 2011.
3. B. Brewington and G. Cybenko. How dynamic is the web? In *WWW '00: Proceedings of the 9th international conference on World Wide Web*, pages 257–276, 2000.
4. A. Brokes, L. Coufal, Z. Flashkova, J. Masanès, J. Oomen, R. Pop, T. Risse, and H. Smulders. Requirement analysis report living web archive. Technical Report FP7-ICT-2007-1, 2008.
5. J. Cho and H. Garcia-Molina. Effective page refresh policies for web crawlers. *ACM Trans. Database Syst.*, 28(4):390–426, 2003.
6. J. Cho and H. Garcia-Molina. Estimating frequency of change. *ACM Trans. Interet Technol.*, 3(3):256–290, 2003.
7. J. Cho, H. Garcia-molina, and L. Page. Efficient crawling through url ordering. In *Computer Networks and ISDN Systems*, pages 161–172, 1998.
8. H. V. de Sompel, M. L. Nelson, R. Sanderson, L. Balakireva, S. Ainsworth, and H. Shankar. Memento: Time travel for the web. *CoRR*, abs/0911.1112, 2009.
9. D. Denev, A. Mazeika, M. Spaniol, and G. Weikum. SHARC: framework for quality-conscious web archiving. *Proc. VLDB Endow.*, 2(1):586–597, 2009.
10. A. Jatowt, Y. Kawai, S. Nakamura, Y. Kidawara, and K. Tanaka. A browser for browsing the past web. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 877–878, New York, NY, USA, 2006.
11. C. Olston and S. Pandey. Recrawl scheduling based on information longevity. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 437–446, New York, NY, USA, 2008.
12. Z. Pehlivan, M. Ben Saad, and S. Gançarski. Vi-diff: Understanding web pages changes. In *21st International Conference on Database and Expert Systems Applications (DEXA'10)*, Bilbao, Spain, 2010.
13. M. Spaniol, D. Denev, A. Mazeika, G. Weikum, and P. Senellart. Data quality in web archiving. In *WICOW '09: Proceedings of the 3rd workshop on Information credibility on the web*, pages 19–26, New York, NY, USA, 2009.
14. M. Spaniol, A. Mazeika, D. Denev, and G. Weikum. "catch me if you can": Visual analysis of coherence defects in web archiving. In *9th International Web Archiving Workshop (IWAW 2009)*, pages 27–37, Corfu, Greece, 2009.
15. J. Teevan, S. T. Dumais, D. J. Liebling, and R. L. Hughes. Changing how people view changes on the web. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 237–246, 2009.