Focused Crawl of Web Archives to Build Event Collections

Focused Crawl of Web Archives to Build Event Collections

Martin Klein
Research Library
Los Alamos National Laboratory
Los Alamos, NM, USA
http://orcid.org/0000-0003-0130-2097
mklein@lanl.gov

Lyudmila Balakireva
Research Library
Los Alamos National Laboratory
Los Alamos, NM, USA
http://orcid.org/0000-0002-3919-3634
ludab@lanl.gov

Herbert Van de Sompel
Research Library
Los Alamos National Laboratory
Los Alamos, NM, USA
http://orcid.org/0000-0002-0715-6126
herbertv@lanl.gov

ABSTRACT

Event collections are frequently built by crawling the live web on the basis of seed URIs nominated by human experts. Focused web crawling is a technique where the crawler is guided by reference content pertaining to the event. Given the dynamic nature of the web and the pace with which topics evolve, the timing of the crawl is a concern for both approaches. We investigate the feasibility of performing focused crawls on the archived web. By utilizing the Memento infrastructure, we obtain resources from 22 web archives that contribute to building event collections. We create collections on four events and compare the relevance of their resources to collections built from crawling the live web as well as from a manually curated collection. Our results show that focused crawling on the archived web can be done and indeed results in highly relevant collections, especially for events that happened further in the past.

1. INTRODUCTION

The pace at which real-world events happen paired with the level of event coverage on the web has by far outgrown the human capacity for information consumption. Therefore, archivists and librarians are interested in building special event-centric web collections that humans can consult post-factum. Web crawling on the basis of seed URIs is a common approach to collect such event-specific web resources. For example, the Archive-It service[1] is frequently used to crawl the web to build archival collections on the basis of seeds URIs[2][3][4] that were manually collected by librarians, archivists, and volunteers. This approach has drawbacks since the notion of relevance is solely based on the nomination of seed URIs by humans. Focused web crawling guided by a set of reference documents that are exemplary of the web resources that should be collected is an approach that is commonly used to build special-purpose collections. It entails an algorithmic assessment of the relevance of the content of a crawled resource rather than a manual selection of URIs to crawl. For both web crawling and focused web crawling, the time between the occurrence of the event and the start of the crawling process is a concern since stories quickly disappear from the top search engine result pages [15], links rot, and content drifts [12, 10]. Web archives around the world routinely collect snapshots of web pages (which we refer to as Mementos) and hence potentially are repositories from which event-specific collections could be gathered some time after the event. However, the various web archives have different scopes e.g., national vs. international resources, cover different time spans, and vary in size of their index[5]. This makes collection building on the basis of distributed web archives difficult when compared to doing so on the live web. Moreover, to the best of our knowledge, focused crawling across web archives has never been attempted. Inspired by previous work by Gossen et al. [9], in this paper, we present a framework to build event-specific collections by focused crawling of web archives. We utilize the Memento protocol [20] and the associated cross-web-archive infrastructure [3] to crawl Mementos in 22 web archives. We build collections by evaluating the content-wise and temporal relevance of crawled resources and we compare the resulting collections with collections created on the basis of live web crawls and a manually curated Archive-It crawl. As such, we take the previous work to the next level and ask

[1] https://archive-it.org/

[2] https://twitter.com/archiveitorg/status/960564121577181184

[3] https://twitter.com/internetarchive/status/806228431474028544

[4] https://twitter.com/internetarchive/status/797263535994613761

[5] https://twitter.com/brewster_kahle/status/954889200083509248

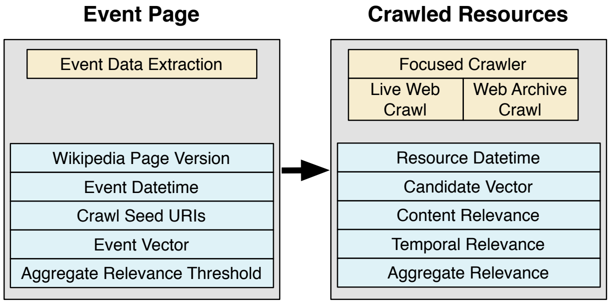


Figure 1: Focused crawling framework

the following questions:

- Can we create event collections by focused crawling web archives?
- How do event collections created from the archived web compare to those created from the live web?
- How does the amount of time passed since the event affect the collections built from the live and the archived web?
- How do event collections built from the archived web compare to manually curated collections?

We consider the main contribution of our work to be the exploration of the feasibility of performing focused crawls on the archived web. To the best of our knowledge, we are the first to do so.

2. RELATED WORK

Previous work by Gossen et al. [9] inspired this work. They developed a focused extraction (not web crawling) system to create event-centric collections from a large static archival collection stored on a server under their control. The content of the Wikipedia page for an event is used to guide the focused extraction. The event datetime is derived from HTML elements in the Wikipedia page and external references in that page are used as seed URIs. They found that their approach outperforms a naive extraction process that is not guided by content and that an approach that combines content-wise and temporal relevance scores mostly performs best. Our approach builds on this work. We deploy a focused crawler that operates on the real web and is not bound to a static, locally stored archival collection. We actually utilize 22 web archives for our crawls and compare the results to comparable focused crawls on the live web. A significant amount of work has been done on focused crawling in general [4, 16, 1]. Some work has additionally explored time-aware focused crawling, such as Pereira et al. [17]. In that work, the authors incorporated temporal data extracted from web pages to satisfy a particular temporal focus of the crawl. They used temporal segmentation of text in a page to determine temporal focus. We follow common practice for our focused crawling approach, for example, by implementing a priority queue. The temporal segmentation of text could have been of interest for our temporal relevance assessment, but, for this experiment we use extraction methods as seen in [6]. Relevant with regard to event-centric collection building is the work by Farag et al. [6] and Littman et al. [14]. Farag et al. introduced an intelligent focused crawling system that works on the basis of an event model. This model captures key data about the event and the focused crawler leverages the event model to predict web page relevance. As a result, the system can successfully filter unrelated content and perform at a high level of precision and recall. The work by Littmann et al. pertains to deriving event collections from social media. The authors focused on increasing the alignment between web archiving tools and processes, and social media data collection practice, for the overall goal of event-centric collection building. Both efforts relate to our work in that the common goal is to build specific collections of web resources. However, both Farag et al. and Littmann et al. are concerned with live web resources only.

3. ESTABLISHING A CRAWLING FRAMEWORK

Our intent is to compare focused crawling of the live web and of web archives for the creation of collections pertaining to unpredictable events such as natural disasters and mass shootings. Inspired by [9], we use the Wikipedia page that describes an event as a starting point. However, we do not use the current version of that page but rather a prior version that is expected to describe the actual event and does not yet include post-event auxiliary content such as references to future related events or analysis of a range of similar events. We select external references of the Wikipedia version page as seeds for crawling and the page's text to assess content relevance of crawled resources. We additionally use a temporal interval starting with the datetime of the event to assess the temporal relevance of crawled resources. For both the live web and web archive crawls, crawled pages that are relevant, both content-wise and temporally, are added to the respective event collection. We describe the details in the remainder of this section and provide a conceptual overview of the framework in Figure 1.

3.1 Wikipedia Page Version

All data required to guide the crawling process is generated from the canonical Wikipedia page of the event. However, our events of interest have happened at some point in the past and their Wikipedia pages, very likely created shortly after the event, have with high probability evolved significantly since then. This raises the question of which version of a Wikipedia page to use as the starting point for our crawls. Since Wikipedia maintains all page versions along with the datetime they were created, we can, in theory, choose any version between the very first and the current one. We know from related work [18] that the majority of edits to a Wikipedia page happen early on in its lifetime. However, event coverage often evolves beyond that point and hence consecutive page edits may still lead to significant changes. For example, other, related events may happen at some later point and may result in the inclusion of new links and references into the event's Wikipedia page. We therefore conjecture that using the current live version of an event's Wikipedia page could introduce too much content and references that do not directly pertain to a description of the event.

We approach the selection of a Wikipedia page version from the perspective of edit frequency. Our goal is to deter-

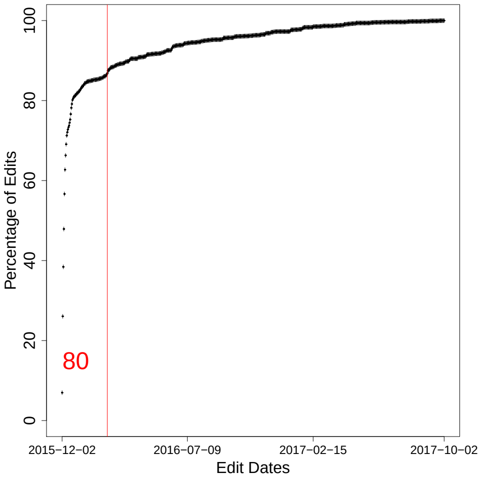


Figure 2: Change point in Wikipedia article edit frequency

mine the date on which the vast majority of edits over the entire history of the page were completed. We select the page with that version date and consider that page to comprehensively capture the essence of the event. For this purpose we plot all edits of a Wikipedia page and their datetimes. Figure 2 shows an example of such a plot where the edit datetimes are on the x-axis and the percentage of edits on the y-axis. We then use the standard R changepoint library [11] to determine the change point in this graph. The change point is the point after which the graph assumes a significantly different shape. In our case, this point is the datetime after which the edit frequency drastically decreases. Hence, we can consider the page version that corresponds with that datetime as capturing the essence of the event. We refer to the change point datetime as DT_{CP}. Figure 2 shows the edits of the San Bernadino Attack Wikipedia page[6] and the detected change point at 80 days after the creation of the page. In this example, we select the version of the Wikipedia page that was live 80 days after the event[7] for our experiment and refer to this version as the DT_{CP} version of the Wikipedia page.

3.2 Event Datetime

The first data point that we extract from the DT_{CP} version of the Wikipedia page is the event datetime. The format and granularity of the provided datetime can vary across Wikipedia pages. For uniformity, we express the event datetime in date, month, year, hour, minute, and seconds. In case no exact time is available from the DT_{CP} version of the Wikipedia page, we set the time to 00:00:01 of the day of the event. We refer to the event datetime as DT_E.

3.3 Crawl Seed URIs

Similar to [9], we extract all external references contained in the DT_{CP} version of the Wikipedia page and consider their URIs as seeds for the focused crawl. For simplicity, we filter out references that do not point to English language content or that point to resources in a representation other than HTML. All remaining references are used as seeds for both the web archive and live web crawls as well as for the content relevance computation outlined below.

3.4 Content Relevance

This section describes the process aimed at determining the extent to which a crawled resource is content-wise relevant for inclusion in the event collection.

3.4.1 Event Vector

We use the textual content of the DT_{CP} version of the Wikipedia page to create an event vector that will serve as our baseline to assess the content relevance of crawled pages. In an effort to stabilize the event vector, we further incorporate the textual content of a random 60% of outgoing references from the DT_{CP} version of the Wikipedia page. In order to reduce noise, such as advertisements, we apply the common boilerpipe library[8], introduced in Kohlschütter et al. [13], to the Wikipedia page as well as to its outgoing references. From the remaining text of the page, we extract 1-grams and 2-grams, store their term frequency (TF), and extract their inverse document frequency (IDF) from the Google NGram dataset [7]. These 1-grams and 2-grams, along with their combined TF-IDF score, make up the event vector.

3.4.2 Candidate Vector and Content Relevance of a Crawled Resource

The textual content of a crawled page is used to generate a candidate vector. We create this candidate vector in a manner very similar to the event vector. After crawling a candidate page, we apply the boilerpipe library and extract the remaining textual content. We determine TF-IDF values from extracted 1-grams and 2-grams to build the candidate vector. We then compute the cosine similarity between the candidate vector and the event vector to obtain a content relevance score R_{cont}. The resulting cosine value is between 0 and 1 where a higher score indicates a greater level of similarity and hence content relevance of the crawled page. The way in which the content relevance is determined is identical for resources in live web and web archive crawls.

3.4.3 Content Relevance Threshold

We compute a content relevance threshold for an event on the assumption that resources referenced in the DT_{CP} version of the Wikipedia page are relevant themselves. We therefore run the same vector computation process for the content of the 40% of references that remain after the process of generating the event vector and compute the cosine similarity between both vectors. We repeat this process 10 times, each time with a different random set of 60% of references for the event vector and hence different remaining 40% of references for comparison. The computed average of the 10 obtained cosine similarity scores serves as our content relevance threshold TH_{cont} for the event.

3.5 Temporal Relevance

This section describes the process aimed at determining the extent to which a crawled resource is temporally relevant for inclusion in the event collection.

[6] https://en.wikipedia.org/wiki/2015_San_Bernardino_attack

[7] https://en.wikipedia.org/w/index.php?title=2015_San_Bernardino_attack&oldid=706012350

[8] https://github.com/kohlschutter/boilerpipe

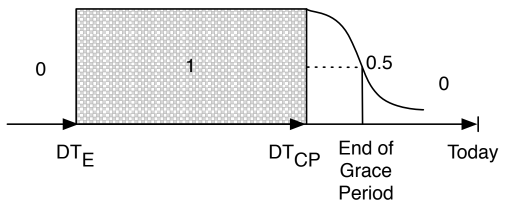


Figure 3: Temporal Relevance Interval

3.5.1 Temporal Interval and Temporal Relevance of a Crawled Resource

Inspired by [9], [8], and [5], we introduce a temporal interval to support assessing whether a crawled resource is temporally relevant. The interval, illustrated by Figure 3, serves the purpose of assigning low temporal relevance score to web resources that were published prior to the event or a long time after it. Equation 1 outlines the computation of the temporal relevance score, which we refer to as R_{temp}. For example, a crawled resource that has an associated datetime DT_R, for example its publication date (see 3.5.2), prior to DT_E gets a temporal relevance score of $R_{temp} = 0$. A resource with DT_R that falls between DT_E and DT_{CP}, on the other hand, is assigned $R_{temp} = 1$. Additionally, a grace period beyond DT_{CP} is considered. The grace period is not unlike the cool-down period introduced in [9] and is additionally motivated by the fact that web archives may take a while to archive a resource after it was published. For web archive crawls, the grace period provides a fair chance for resources that were published some time before DT_{CP} but archived beyond it to still be considered relevant. As can be seen in Equation 1, during the grace period, a resource can obtain a R_{temp} score of less than 1 and greater or equal to 0.5. In this equation, $\Delta t\prime$ represents the difference between DT_{CP} and DT_R and Δt is equal to 1/4 of the period between DT_E and DT_{CP}. Different arguments can be made regarding the choice of the length of the grace period. Rather than setting a duration arbitrarily, we determine it using the time it took for references in the DT_{CP} version of the Wikipedia page to be archived. More specifically, we use the average time between the datetimes associated with all references of the DT_{CP} version of the Wikipedia page (as indicated in the article) and their corresponding archival datetime as the length of the grace period. For resources captured in the live crawl, we apply a grace period to give certain resources published past DT_{CP} a fair chance to be considered relevant. In this case, we determine its duration as the average distance between the associated datetimes of all references from the DT_{CP} version of the Wikipedia page (as indicated in the article).

$$R_{temp} = \begin{cases} 1 & \text{if } DT_E \leq DT_R \leq DT_{CP} \\ 0 & \text{if } DT_E > DT_R \\ e^{-\left(\left(\frac{ln(2)}{\Delta t}\right) \, * \, \Delta t\prime\right)} & \text{if } DT_R > DT_{CP} \end{cases} \tag{1}$$

3.5.2 Resource Datetime

As described in the previous section, the datetime DT_R associated with a crawled resource plays a core role in determining its temporal relevance score. The manner in which this datetime is obtained is different for live web resources and Mementos. To determine the DT_R for a resource from the live web crawl, we use various approaches, some of which have also been used in Farag et al. [6]. The first approach is to extract a datetime from the URI of a page, as many news publishers use URI patterns that contain a datetime, for example: http://www.cnn.com/2017/12/09/us/wildfire-fighting-tactics/. Second, we consider the page's HTML, as news publishers and content management systems frequently embed datetimes. For example, the following HTML excerpt is from a New York Times article:

```
<meta property="article:published"
      itemprop="datePublished"
      content="2017-12-09T10:14:50-05:00"/>
```

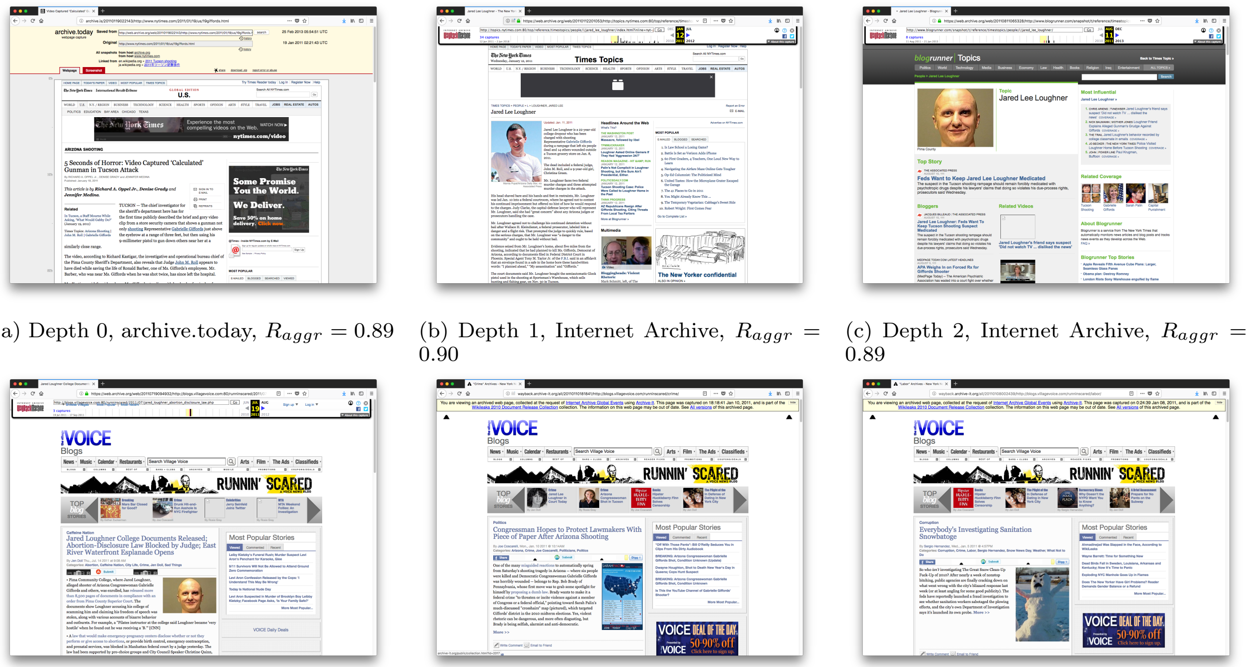
Third, we utilize the CarbonDate tool[9], first introduced by SalahEldeen and Nelson [19]. The tool looks for first mentions of the URI on Twitter and Bitly. If these methods return more than one extracted datetime, we choose the earliest one as the page's DT_R. If all methods fail and no datetime is extracted, we dismiss the crawled resource.

To determine the DT_R for a resource from the archived web crawl, a feature of the Memento protocol [20] that is supported by all archives included in the experiment, is leveraged because it yields a datetime with minimal effort involved. A web archive that returns a Memento also returns the datetime it was archived in the `Memento-Datetime` HTTP response header. If this datetime falls within the temporal interval between DT_E and DT_{CP} as shown in Figure 3, we use it as our DT_R. This kind of Memento will obtain a temporal relevance score of 1. Understanding that web archives commonly archive pages quite some time after they were published, this approach can lead to pages that were published prior to DT_E (but archived past it) receiving a score of 1. However, given the temporal threshold will be combined with a content-based threshold, this risk is outweighed by the benefit of a straightforward means to determine a DT_R. In cases where the archival datetime is beyond DT_{CP}, we can not merely dismiss the Memento because it could have been archived a long time after it was initially published. Hence, in these cases we attempt to determine the publication date of the page on the basis of the Memento. To that end, we use the CarbonDate tool again. If the tool can assign a date to the Memento, we use it as DT_R. If the tool is unsuccessful, we leverage archived HTTP headers, which some web archives convey as custom X-headers in the HTTP response of a Memento. For example, if a Memento provides an `X-Last-Modified` header, we use its datetime as DT_R. If all methods fail, the crawled resource is dismissed.

3.5.3 Temporal Relevance Threshold

We compute a temporal relevance threshold for an event on the assumption that resources referenced in the DT_{CP} version of the Wikipedia page are temporally relevant themselves. We therefore compute the temporal relevance of each URI in the same random set of 60% of references that we use for the computation of TH_{cont}. We repeat this process 10 times, each time with a different set of random 60% and use the computed average of all obtained scores as our temporal relevance threshold TH_{temp}.

[9] http://carbondate.cs.odu.edu/



(a) Depth 0, archive.today, $R_{aggr} = 0.89$

(b) Depth 1, Internet Archive, $R_{aggr} = 0.90$

(c) Depth 2, Internet Archive, $R_{aggr} = 0.89$

(d) Depth 3, Internet Archive, $R_{aggr} = 0.89$

(e) Depth 4, Archive-It, $R_{aggr} = 0.91$

(f) Depth 5, Archive-It, $R_{aggr} = 0.51$

Figure 4: Mementos resulting from the TUC web archive crawl at depth 0 (seed) through depth 5 obtained from various web archives using the Memento infrastructure

3.6 Aggregate Relevance and Aggregate Relevance Threshold

Following the same reasoning as in [9], we use an aggregate relevance score R_{aggr} based on the sum of the content and temporal relevance scores, respectively R_{cont} and R_{temp}. In order to aggregate both scores into one, we introduce two weighting factors α and β, as shown in Equation 2. These factors can be used to weigh the significance of either relevance score. For our experiments we balance the weight equally and assign the value of 0.5 to both α and β, as also seen in [9].

$$R_{aggr} = \alpha * R_{cont} + \beta * R_{temp} \qquad (2)$$

$$TH_{aggr} = \alpha * TH_{cont} + \beta * TH_{temp} \qquad (3)$$

Similarly, as shown in Equation 3, we define an aggregate threshold. We use the same weighting factors as seen in Equation 2 to balance the significance of both parts.

Based on the R_{aggr} score of a page and the computed TH_{aggr} of the corresponding event, we determine whether the crawled page will be selected for the event collection or not. We classify a page with an aggregate relevance score equal to or above the threshold ($R_{aggr} \geq TH_{aggr}$) as relevant and hence select it into the collection. On the other hand, we consider a page with a score below the threshold ($R_{aggr} < TH_{aggr}$) as not relevant and reject it.

4. CRAWLING THE LIVE AND ARCHIVED WEB

Our crawling process, just like other implementations of focused crawlers, is deployed with a priority queue that informs the crawler which URIs to crawl next. In our case, resources linked from pages with a higher aggregate relevance score will be ranked higher in the priority queue. Our crawling process also needs to stop at some point. The simplest stop condition for a focused crawler is when the queue is empty and there are no documents left to crawl. However, under this condition, depending on the event and the length of the list of seed URIs, the crawl can run for a long time. Other typical stop conditions for crawlers are a maximum number of documents crawled, a maximum size of the crawled dataset, a maximum runtime, or a maximum crawl depth. We chose to implement the latter condition and run our focused crawler for a maximum depth of six. A seed URI is considered depth 0 and as long as the outlinks remain relevant, our crawler follows outlinks up until crawl depth 5. Arguably, the chosen crawl depth is somewhat arbitrary but our preliminary tests indicated that smaller depths tended to result in too few documents and larger depths took too long to complete. Clearly, this stop condition is configurable and we leave a thorough investigation of an optimal stop condition for future work. We modify the code base of the `crawler4j`[10] tool for our focused crawler and run all crawls on an Amazon virtual machine.

The remainder of this section provides further details about the crawling process with a focus on web archive crawling because, to the best of our understanding, the work described here is the first to use focused crawling across web archives.

4.1 Live Web Crawls

The crawl of the live web follows established focused crawl

[10] https://github.com/yasserg/crawler4j

Table 1: Crawled events

Event	DT_E	DT_{CP}	**Wikipedia page version**
NYC	10/31/2017	NA	https://en.wikipedia.org/wiki/2017_New_York_City_truck_attack
SB	12/02/2015	02/20/2016	https://en.wikipedia.org/w/index.php?title=2015_San_Bernardino_attack&oldid=706012350
TUC	01/08/2011	01/12/2012	https://en.wikipedia.org/w/index.php?title=2011_Tucson_shooting&oldid=471037980
BIN	04/03/2009	11/11/2009	https://en.wikipedia.org/w/index.php?title=Binghamton_shootings&oldid=325176468

practice, starting by fetching a seed URI page from the live web and determining and evaluating its aggregate relevance R_{aggr}. If the page is deemed relevant, it is added to the event collection, its outlinks are extracted and added to the priority queue. Each URI in the priority queue is handled in the same manner until the crawler's stop condition is met.

4.2 Web Archive Crawls

Crawling the archived web is done by utilizing the Memento protocol [20] and associated infrastructure. Unlike previous work [9], in order to generate the richest possible event collections, we are interested in obtaining Mementos from as many publicly available web archives around the world as possible. The Memento infrastructure, and in particular the Memento Aggregator [3], makes this possible. For each URI that needs to be crawled (seed URIs and URIs in the priority queue) until the crawler's stop condition is met, the crawler obtains a Memento of that URI that was archived temporally closest but after DT_E. Closest to that datetime, in order to avoid using a version of the resource for which the content may have drifted [10] since it was originally linked to. And after that datetime because, clearly, pages that were archived prior to DT_E were also published before it and hence are not relevant when unplanned events are concerned.

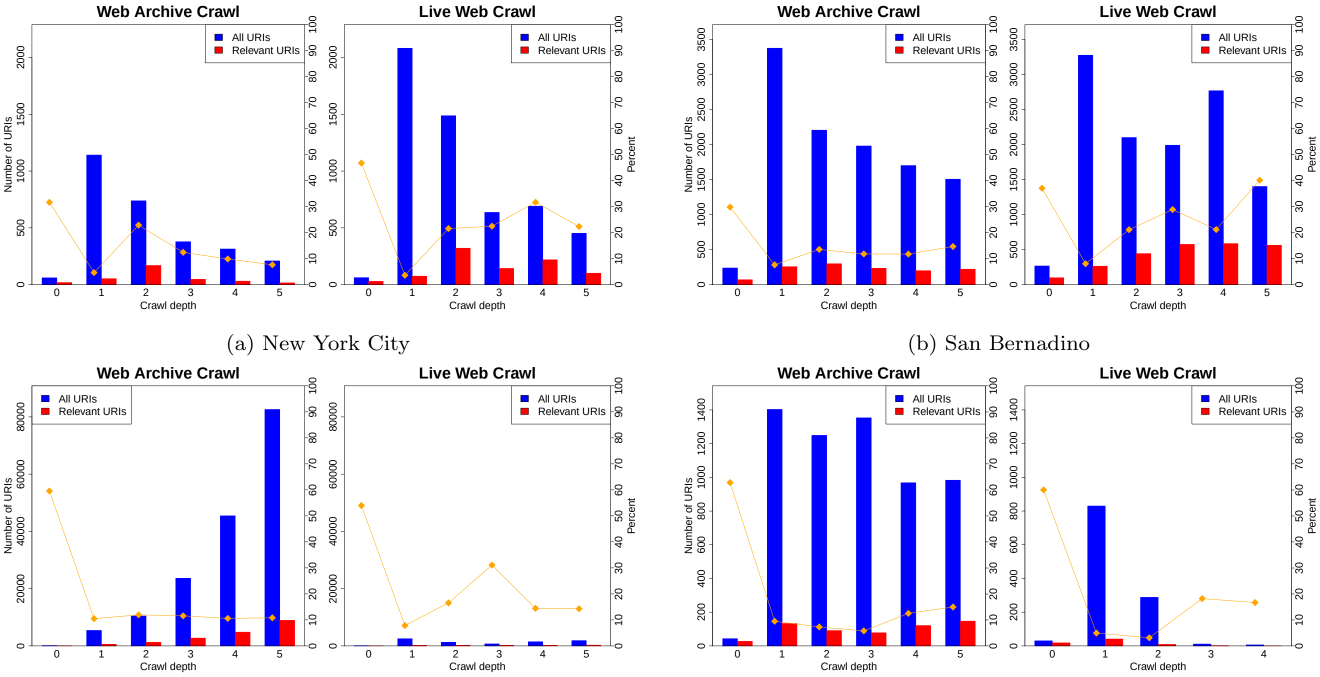
The Memento protocol and the Memento Aggregator provide two ways to discover a Memento with an archival date closest to a desired date. The TimeMap approach consists of requesting a list of URIs of all available Mementos (URI-Ms in Memento protocol lingo) for a certain original URI (URI-R in Memento protocol lingo). From that list, the Memento closest to and after DT_E can be selected. The TimeGate approach entails performing datetime negotiation by providing an original URI as well as a preferred archival datetime, and receiving the URI of the Memento with an archival datetime temporally closest to the preferred datetime in return. However, this approach can yield a Memento that is either prior to or after the preferred datetime. Both the TimeMap and TimeGate approaches require the Memento Aggregator to issue a request to multiple web archives for each URI. As such, in both cases, extra HTTP requests are involved when compared to live web crawling where a URI is accessed directly. Therefore, a web archive crawl will necessarily be slower than a live web crawl. However, the TimeMap approach can involve significantly more HTTP requests than the TimeGate approach because obtaining a complete TimeMap from a single archive itself may entail multiple requests. As such, in order to reduce the overall web archive crawling time, we use the TimeGate approach for our experiments and use DT_E as the preferred datetime. In case the returned Memento has an archival datetime prior to our DT_E, we simply follow the `next memento` HTTP link header, which is provided in the TimeGate HTTP response. This header points to the temporally "next" Memento that, as per the Memento protocol's datetime negotiation, has a datetime greater or equal to DT_E.

For each URI that needs to be crawled, this process yields the URI of a Memento. The crawler fetches that Memento from the web archive that holds it, computes its R_{aggr} score and evaluates it vis-a-vis the TH_{aggr}. If the Memento is deemed relevant, it is added to the event collection, its outlinks are extracted and added to the priority queue. We note that most web archives rewrite outlinks in their Mementos to point back into the same archive rather than to the live web, even when the archive does not hold a Memento for the linked resource or only holds Mementos that are temporally distant from the desired time, which in our case is DT_E [2]. We therefore add the original URI (URI-R) of the outlink, which can be obtained using features of the Memento protocol, to the priority queue rather than the rewritten URI-M of the outlink. This allows us to discover the Memento for outlinks that is temporally closest and past the event datetime DT_E across all web archives covered by the Aggregator.

Figure 4 shows six screenshots of consecutively crawled Mementos. Figure 4a shows the Memento of the seed URI, Figure 4b the Memento of one of the seed's outlinks (crawl depth 1), Figure 4c the Memento of an outlink of the prior Memento of crawl depth 1 (crawl depth 2), and so on. These screenshots show the diversity of contributing web archives: the Memento for the seed URI was found in `archive.today`, the Mementos for crawl depths 1..3 were provided by the Internet Archive, and depths 4..5 by Archive-It. The figure also shows the R_{aggr} scores for each Memento. The threshold TH_{aggr} for this crawl was 0.75 and hence the first five Mementos are classified as relevant but the last one is not. Since our crawl depth was set to five, the outlinks of the Memento shown in Figure 4f were not added to the priority queue. If, however, it had been set to a number larger than five, the Memento's outlinks would also not have been added to the queue as the Memento's $R_{aggr} < TH_{aggr}$.

5. WEB CRAWL COMPARISON

We present the results of crawls for four different events: the 2017 New York City attack (NYC), the 2015 San Bernadino attack (SB), the 2011 Tucson shooting (TUC), and the 2009 Binghampton shootings (BIN). We chose these events because they are fairly similar in nature, they all happened in the U.S., and their coverage on the web is predominantly in English. We assumed that this uniformity would better support detecting patterns in our results. We ran our crawls in November of 2017, a few days after the New York City attack, and more than eight years after the Binghampton shootings. Table 1 summarizes the four events for which we created an event collection with our focused crawling framework. The table also shows the event dates DT_E, the change points DT_{CP}, and the URIs of the DT_{CP} versions of the Wikipedia event page. Note that we did not compute a change point for the NYC event because we crawled re-



(a) New York City

(b) San Bernadino

(c) Tucson

(d) Binghampton

Figure 5: Relevant URIs

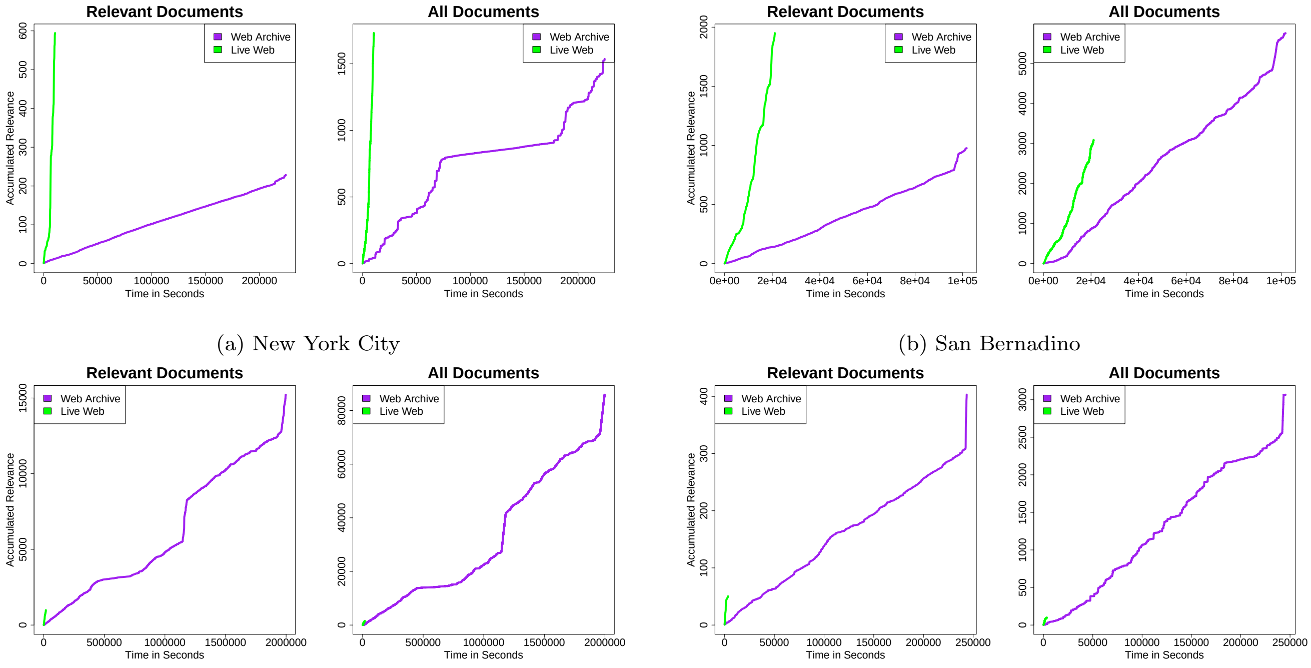
sources very soon after the attack happened, at which point the number of Wikipedia page edits had not yet reached the change point. As such, for the NYC event, we used the live version of the Wikipedia event page as it was at the time of crawling.

5.1 Relevant URIs

Our first results are visualized in Figure 5, distinguished by event. For example, the crawl data for the New York City attack is shown in Figure 5a, for the San Bernadino attack in Figure 5b, and so on. The left-hand plot for each event shows the results from the web archive crawl, and the right plot displays our results from the live web crawl. All subfigures of Figure 5 show the number of URIs crawled at each crawl depth (0..5). The blue bars indicate the total number of URIs crawled and the red bars represent the number of URIs that were classified as relevant, per corresponding crawl depth. The bars refer to the left y-axis. The lines, representing the fraction of relevant URIs, refer to the right y-axis. For the NYC event, the live web crawl is the clear winner as it returns significantly more URIs as well as relevant URIs. The fraction of relevant URIs on depth 0 (the seeds) is almost 50% for the live web vs. 30% for the web archive crawl. On crawl depth 1, the first outlinks from the seeds, and on depth 2, the fractions are fairly similar. But for the further depths 3, 4, and 5 the live crawl shows ratios above 20% of relevant URIs whereas the web archive crawl only shows ratios around 10%. This result makes intuitive sense as we conducted the crawl merely days after the event happened. It is highly likely that web archives did not have a chance to archive a significant amount of the relevant resources and hence our web archive crawl did not surface many (relevant) URIs. The results for the SB crawls, shown in Figure 5b, are similar in that the live crawl returns a higher ratio of relevant URIs at all crawl depths. While the number of total URIs crawled is comparable between both crawls, the number of relevant URIs is consistently higher for the live web crawl. Our interpretation of these results is that, since the event datetime is two years in the past, web archives have had enough time to create Mementos of many relevant web pages. However the web archive crawl does not outperform the live web crawl. Figures 5c and 5d show a very different pattern. In both cases the live web crawl results in fewer total URIs and fewer relevant URIs crawled than the web archive crawl. The BIN live crawl does not even return any URIs on depth 5. Our interpretation of this pattern is based on the fact that the TUC and BIN events happened in 2011 and 2009, respectively. Hence, a lot of time has passed for pages on the live web to either completely disappear or to have their content drift to something less relevant compared to the event vector. This is a phenomenon that we have previously investigated in the realm of scholarly communication [12, 10] and that seems to also happen for web coverage of unplanned events. In essence, our finding suggests that live web resources pertaining to an event that were available at the time of the event are by now more likely available in web archives than on the live web.

5.2 Accumulated Relevance

Inspired by the evaluation shown in [9], we also analyze the accumulated relevance of all crawled resources, understanding that even resources that do not meet the aggregate relevance threshold still have an aggregate relevance score. Just like in this related work, we simply add individual R_{aggr} scores of all crawled resources to obtain the accumulated relevance. Since our crawl stop condition is defined by crawl depth, we are able to show two different analyses of the accumulated relevance. First, we present the



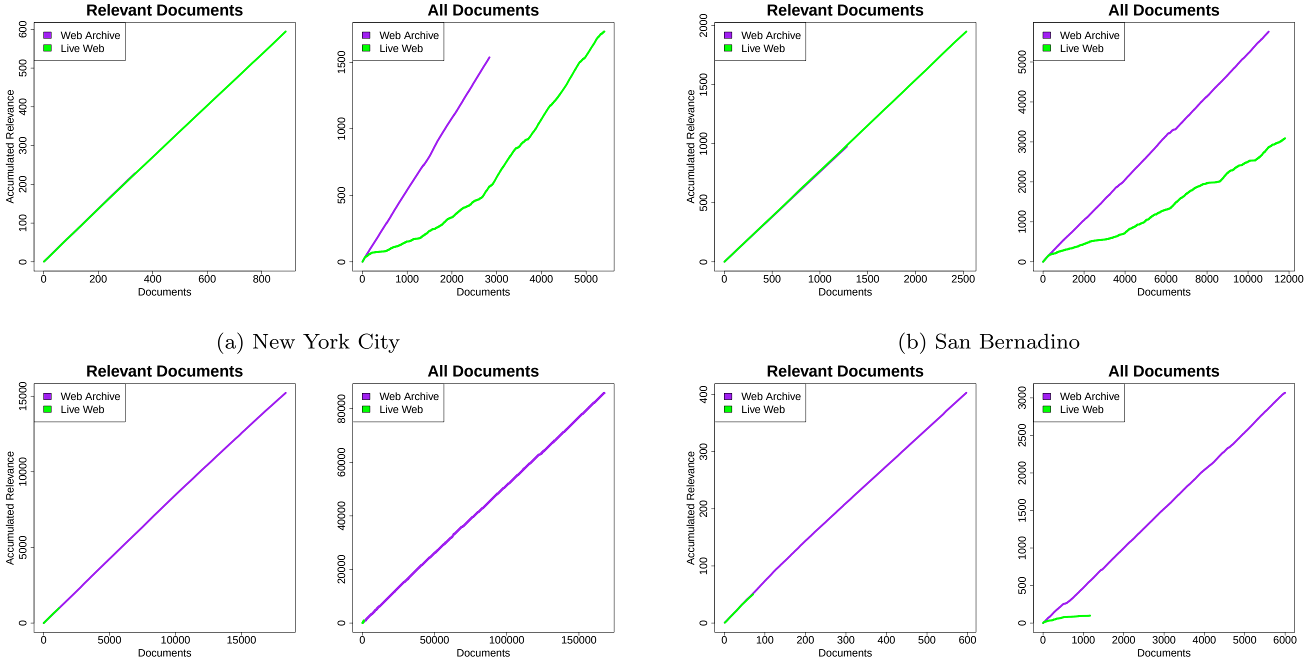
(a) New York City

(b) San Bernadino

(c) Tucson

(d) Binghampton

Figure 6: Accumulated relevance over time



(a) New York City

(b) San Bernadino

(c) Tucson

(d) Binghampton

Figure 7: Accumulated relevance over documents

accumulated relevance over elapsed crawl time. We expect the web archive crawl to take longer than the live crawl as we query the Memento Aggregator for each candidate URI. As described earlier, this results in polling several of the 22 compliant web archives, which adds to crawling time.

Figure 6 displays the accumulated relevance (on the y-axis) over time (on the x-axis) for all four events. The green lines represent the live web crawl and the purple line the web archive crawl. Each subfigure shows two distinct plots. The plot on the left-hand side shows the data for all re-

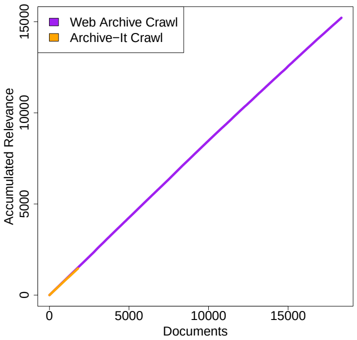


Figure 8: TUC web archive crawl vs Archive-It crawl

sources that were classified as relevant. The plot on the right shows the data for all crawled resources, including the ones that were crawled because their parent was categorized as relevant but they themselves had a relevance score below the threshold. These resources, while "failing" our threshold test, may still have value for an event-centric collection and hence they are considered here. Unlike the previous figures, subfigures of Figure 6 do not distinguish between crawl depths.

Figure 6a shows the accumulated relevance over time for the NYC crawl. Considering all relevant documents (plot on the left), we can observe that the accumulated relevance of the live crawl increases very rapidly and that the web archive crawl takes much longer, as expected, and never reaches the same relevance. The accumulated relevance for all crawled documents (plot on the right) for the web archive crawl gets closer but still does not reach the accumulated relevance level of the live crawl. Given the results from the previous section, these observations are not surprising.

Figure 6b shows a similar picture for relevant documents in the SB crawl. However, the data for all crawled documents is surprising. The web archive crawl takes much longer but eventually surpasses the accumulated relevance level of the live web crawl. Figures 6c and 6d show an even more dramatic picture. The accumulated relevance of the live web crawls is quickly surpassed by the web archive crawls. Given relatively few URIs were obtained in the live crawls (as seen in Figures 5c and 5d), it is not surprising to see these crawls finish rather quickly. The web archive crawls, again, take significantly longer to complete.

The second analysis of the accumulated relevance is over the number of documents crawled. Figure 7 visualizes this data in a similar fashion as seen in the previous figures. The data for the NYC crawl is displayed in Figure 7a where we see twice as many relevant documents for the live web than for the web archive crawl. The accumulated relevance therefore is much higher. When we consider all crawled documents, we also find roughly twice as many resources in the live crawl and, while the relevance of the web archive crawl is closer, it does not catch up. These data points confirm our previous findings.

The picture for the live crawl of the SB event (Figure 7b) is similar to the NYC event. The plot for all documents, however, shows an interesting fact: the total number of documents crawled is very similar (11,806 for the live web vs. 11,007 for the web archive crawl) while the accumulated relevance of the web archive crawl ends up to be almost twice that of the live web crawl. As previously indicated in Figures 6c and 6d, Figures 7c and 7d confirm that web archive crawls perform considerably better than live crawls for the TUC and BIN events, respectively.

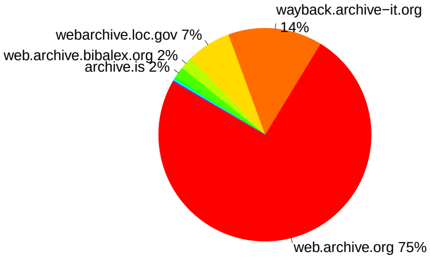


Figure 9: Contributions to the TUC web archive crawl

6. COMPARISON TO A MANUALLY CREATED COLLECTION

We utilized Wikipedia event pages, specifically the URIs of external references as seeds for our crawls. However, a common approach for building event-centric collections from web pages is based on manual suggestion of seed URIs. We are therefore motivated to compare our approach with an event collection that was created using manually selected seed URIs. The Archive-It service provided by the Internet Archive is frequently used to build such collections. At the time we conducted our experiments, the only Archive-It collection that matched one of our events was the Tucson shooting collection, originally created by scholars at Virginia Tech. We were able to obtain a copy of the crawled data and compared it to our TUC web archive crawl.

To build this collection, the Archive-It crawler was configured to merely crawl all 1,997 seed URIs and not go beyond this crawl depth. In terms of our experiment, this equals to crawl depth 0 and hence a comparison of relevant URIs per crawl depth 0..5 is not applicable. Instead, we compute the accumulated relevance of all crawled resources and compare it to the data from our web archive crawl. Figure 8 shows the results. It is apparent that the Archive-It crawl has significantly fewer documents crawled compared to our web archive crawl, an obvious result of the crawl depth constraint. However, what is interesting is that the slope of the line is equally steep for both crawls i.e., the orange line (Archive-It crawl) and the purple line (our web archive crawl). It would not have been unreasonable to assume that the manually curated seed list would result in more relevant URIs crawled than the automatically generated seed list stemming from the references of the DT_{CP} version of the Wikipedia page.

This comparison raises the question of the level of overlap between the manually curated URIs from the Archive-It collection and the automatically crawled URIs of our TUC web archive crawl. We classified 1,795 out of all 1,997 URIs in the Archive-It collection as relevant. On the other hand, we deemed 18,353 out of 167,641 crawled URIs in the TUC archived crawl relevant. We found that only 92 URIs overlap in both collections, which indicates that both collections are rather disjoint.

Another distinguishing element between these two crawls is the variety of web archives that contribute to the crawl. Given our framework for crawling the archived web, we are able to crawl archived resources from a total of 22 web archives. Naturally, the Archive-It crawl only stems from one archive. Figure 9 shows the distribution of web archives that have contributed to our TUC archived crawl. The figure shows the top five contributing archives only, with the Internet Archive providing 75% of all Mementos. We note, however, the diversity of other contributing archives. Besides resources provided from the Library of Congress and the Library of Alexandria, as shown in Figure 9, our crawl further includes resources crawled from the Portuguese, the Icelandic, the UK, and the Northern Ireland Web Archives, not labeled in Figure 9.

7. CONCLUSION AND FUTURE WORK

Inspired by previous work, we were motivated to investigate a focused crawling approach to build event-centric collections. In this paper we outline our focused crawling framework, detail its methodology, describe its crawling process of the live and archived web, and present the results on four unpredictable events. Our results prove that focused crawling on the archived web is feasible. The Memento protocol and infrastructure play a vital role in this process.

Comparing web archive crawls and live web crawls for events, we observe the following patterns:

1. For rather recent events, such as the NYC event in our experiments, a crawl of the live web results in more total URIs, more relevant URIs, and a higher level of accumulated relevance over all documents. A web archive crawl is not competitive and takes much longer to complete.

2. For events that are less recent but took place in the not too distant past, such as the SAN event in our experiments, our results show a mixed pattern. If we consider relevant documents only, the live web crawl outperforms the web archive crawl and, as expected, finishes much quicker. However, if time is not a main concern and we can consider all crawled resources, the web archive crawl provides more documents that, in aggregate, are more relevant.

3. For events that happened in the more distant past, such as the TUC and BIN events in our experiments, the web archive crawl, while taking much longer to complete, returns many more relevant results. A live web crawl does not provide compelling results.

The comparison of our web archive crawl on the TUC event with the manually curated Archive-It crawl shows that both collections, while distinct in terms of their crawled URIs, are highly relevant to the event. In addition, we find that the inclusion of an array of web archives clearly provides merit to the collection building. We therefore suggest that, especially for collections of events that took place in the more distant past, augmenting manually curated collections that are based on human-evaluated seed URIs with a focused crawl that is based on the extraction of references from Wikipedia pages can be very beneficial.

Our chosen events are constrained in dimensions such as event type, language, location and hence more experimentation is required to draw general conclusions from our findings. In addition, various aspects of our crawling framework (event vector, threshold computation, weighting factors) deserve further evaluation in the future.

8. ACKNOWLEDGMENTS

We would like to express our thanks to Liuqing Li, Edward Fox, and Zhiwu Xie from Virginia Tech for making their Archive-It collection on the Tucson shooting available for this experiment.

9. REFERENCES

[1] C. C. Aggarwal, F. Al-Garawi, and P. S. Yu. Intelligent crawling on the world wide web with arbitrary predicates. In *Proceedings of WWW'01*, pages 96–105, 2001.

[2] S. G. Ainsworth and M. L. Nelson. Evaluating sliding and sticky target policies by measuring temporal drift in acyclic walks through a web archive. *International Journal on Digital Libraries*, 16(2):129–144, 2015.

[3] N. J. Bornand, L. Balakireva, and H. Van de Sompel. Routing memento requests using binary classifiers. *CoRR*, abs/1606.09136, 2016.

[4] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31(11):1623 – 1640, 1999.

[5] M. Costa, F. Couto, and M. Silva. Learning temporal-dependent ranking models. In *Proceedings of SIGIR '14*, pages 757–766, 2014.

[6] M. M. G. Farag, S. Lee, and E. A. Fox. Focused crawler for events. *International Journal on Digital Libraries*, 19(1):3–19, 2018.

[7] Y. Goldberg and J. Orwant. A dataset of syntactic-ngrams over time from a very large corpus of english books. In *Proceedings of *SEM' 13*, volume 1, pages 241–247, 2013.

[8] G. Gossen, E. Demidova, and T. Risse. iCrawl: Improving the Freshness of Web Collections by Integrating Social Web and Focused Web Crawling. In *Proceedings of JCDL '15*, pages 75–84, 2015.

[9] G. Gossen, E. Demidova, and T. Risse. Extracting event-centric document collections from large-scale web archives. In *Proceedings of TPDL' 17*, pages 116–127, 2017.

[10] S. M. Jones, H. Van de Sompel, H. Shankar, M. Klein, R. Tobin, and C. Grover. Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. *PLoS ONE*, 11(12), 2016.

[11] R. Killick and I. A. Eckley. changepoint: An R package for changepoint analysis. *Journal of Statistical Software*, 58(3):1–19, 2014.

[12] M. Klein, H. Van de Sompel, R. Sanderson, H. Shankar, L. Balakireva, K. Zhou, and R. Tobin. Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot. *PLoS ONE*, 9(12), 2014.

[13] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of WSDM '10*, pages 441–450, 2010.

[14] J. Littman, D. Chudnov, D. Kerchner, C. Peterson, Y. Tan, R. Trent, R. Vij, and L. Wrubel. Api-based

social media collecting as a form of web archiving. *International Journal on Digital Libraries*, 19(1):21–38, 2018.

[15] A. C. Nwala, M. C. Weigle, and M. L. Nelson. Scraping serps for archival seeds: it matters when you start. In *Proceedings of JCDL'18*, 2018.

[16] G. Pant and P. Srinivasan. Learning to crawl: Comparing classification schemes. *ACM Transactions on Information Systems (TOIS)*, 23(4):430–462, 2005.

[17] P. Pereira, J. Macedo, O. Craveiro, and H. Madeira. Time-aware focused web crawling. In *European Conference on Information Retrieval*, pages 534–539, 2014.

[18] J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani. Characterizing and modeling the dynamics of online popularity. *Phys. Rev. Lett.*, 105:158701, 2010.

[19] H. SalahEldeen and M. L. Nelson. Carbon dating the web: Estimating the age of web resources. *CoRR*, abs/1304.5213, 2013.

[20] H. Van de Sompel, M. L. Nelson, and R. Sanderson. HTTP Framework for Time-Based Access to Resource States – Memento, 2013. https://tools.ietf.org/html/rfc7089.