Extracting Event-Centric Document Collections from Large-Scale Web Archives

Gerhard Gossen, Elena Demidova, and Thomas Risse

L3S Research Center, Leibniz Universität Hannover
{gossen, demidova, risse}@l3s.de

Abstract. Web archives are typically very broad in scope and extremely large in scale. This makes data analysis appear daunting, especially for non-computer scientists. These collections constitute an increasingly important source for researchers in the social sciences, the historical sciences and journalists interested in studying past events. However, there are currently no access methods that help users to efficiently access information, in particular about specific events, beyond the retrieval of individual disconnected documents. Therefore we propose a novel method to extract event-centric document collections from large scale Web archives. This method relies on a specialized focused extraction algorithm. Our experiments on the German Web archive (covering a time period of 19 years) demonstrate that our method enables the extraction of event-centric collections for different event types.

1 Introduction

Web archives created by the Internet Archive[1] (IA), national libraries and other archiving services contain large amounts of information collected for a time period of over twenty years [6]. These archives constitute a valuable source for research in many disciplines, including the digital humanities, the historical sciences and journalism by offering a unique possibility to look into past events and their representation on the Web. They can enable a better understanding of past events and offer a lot of novel research directions for these disciplines.

Most Web archive services aim to capture the entire Web (IA) or national top-level domains (national libraries) and are therefore very broad in their scope. Consequently they are also very diverse regarding the topics they contain and the time intervals they cover. Due to the large size and the broad scope it is difficult for interested researchers to locate relevant information in the archives as search facilities are very limited compared to the live Web.

In previous work [26,14] we have argued that these users are typically interested in studying smaller and more focused event-centric collections of documents contained in a Web archive. Such collections can reflect specific events such as elections, sports tournaments or natural disasters, for example the Fukushima

[1] https://archive.org

nuclear disaster in 2011, the German federal election in 2009 or the FIFA World Cup 2006, especially in regard to their media coverage and public reactions.

Archive services such as Archive-IT[2] collect documents around specific events. These *special collections* are however defined and crawled on an individual basis, such that users are restricted to the collections that exist and their selected scope. Other existing access methods to temporal Web collections do not support creating ad-hoc collections, often forcing users to create their own corpora manually. Currently, access to large-scale Web archives is limited to browsing of individual Web pages through browser-based tools such as the Wayback machine[3], or initial support for keyword-based access[4]. However, these access methods are not sufficient for several reasons. First, the Wayback machine requires the user to already know the URL of the document. Second, full-text indexing of large-scale archived collections incurs high processing and storage costs. Third, such indexes only allow retrieval of individual disconnected documents. Instead, automatic methods are needed that can extract collections of documents related to a particular event of user interest. These collections need to preserve the original link structure to achieve a high degree of authenticity and enable the application of analytical methods on the relevant parts of the Web archive [14].

In this paper, we present a starting point for tackling the novel problem of extracting topically and temporally coherent, interlinked event-centric document collections from large-scale and broad scope Web archives. The key contributions of this paper are: (1) a definition of a *Collection Specification* that describes the temporal and topical scope of the collection to be extracted and gives the user intuitive but powerful options to control the data collection process; and (2) a *focused crawling-based extraction method* for Web archives to create event-centric collections without requiring any full-text indexes. We evaluate our approach in a local environment using file system crawling. However, our approach can easily be used across Web archives using existing access methods. We make our source code and evaluation data available to encourage further research[5].

2 Related Work

Our method is related to crawling methods for creating Web Archives (e.g. [17,24]), as well as to methods for temporal information retrieval [5].

The collection of Web documents from the live Web for retrieval and archiving purposes is usually performed using Web crawlers. Crawling methods that aim to create broad scope collections for search and archiving purposes intend to capture as much of the Web as possible. An example of a web-scale archiving crawler currently used by the Internet Archive is Heritrix [20]. In contrast, *focused crawling* [4] aims to only collect pages that are related to a specific topic. Focused crawlers [1,22] learn a model of the topic and follow links only if they are

[2] `https://archive-it.org/`
[3] `http://netpreserve.org/openwayback`
[4] `https://blog.archive.org/2016/10/24/beta-wayback-machine-now-with-site-search/`
[5] `https://github.com/gerhardgossen/archive-recrawling`

Table 1. Examples of temporal event characteristics.

Event	Type	Duration	Lead time	Cool-down time
Olympic games	recurring	2 weeks	weeks	days
Federal election	recurring	1 day	months	weeks
Fukushima accident	non-recurring	1 week	—	months
Snowden leaks	non-recurring	1 day	—	years

expected to match that topic, e.g. based on the page containing the link. This follows the obervation that relevant documents will preferentially link to other relevant documents ("topical locality" [1]). Extensions of this model use ontologies to incorporate semantic knowledge into the matching process [10,9], 'tunnel' between disjoint page clusters [3,25] or learn navigation structures necessary to find relevant pages [7,17]. In time-aware focused crawling [24] the document or event time is used as the primary focusing criterion. In event-based crawling [11] events are described using an event model that incorporates event location and date. Here Web page relevance is computed as a weighted average of content, location and date similarity. As location extraction increases the overall complexity of the process, we focus on the content and time-based features. Freshness as a specific aspect of temporal relevance has been addressed in the context of joint crawling of the Web and Social media sites [12] where URLs present in Social media posts are used as entry points to recently published content on the Web. In summary, most existing approaches to focused Web crawling consider the topical and temporal relevance in isolation and do not address the problem of jointly finding temporally and topically relevant content. Furthermore, whereas existing approaches operate on the live Web, we are the first to apply focused crawling techniques to existing Web archives.

The notion of temporal relevance has also been explored in the area of temporal information retrieval. Existing ranking methods have been extended to rank documents based on their creation time [5] or to diversify search results over relevant time periods [2]. Contemporary search engines also rank documents based on their freshness (estimated based on their crawling history) [8]. Similarly, time information has been combined with the hypertext link graph to detect the most relevant documents for a given query [21]. These approaches depend on full-text or graph indexes and therefore have a high up-front computational and index storage cost. Moreover, these approaches only allow retrieval of individual disconnected documents and do not preserve the link structure. In contrast, our method allows on demand extraction of interlinked event-centric collections without requiring any additional indexes on the archive.

3 Event-Centric Collections

Events are typically characterized through a certain date or a time interval such as the date of an accident or the duration of a tournament. Here the event time

interval is clearly defined. Nevertheless, event-related documents also appear outside of this time interval. For planned and in particular regularly recurring events such as sports competitions or elections, relevant documents often appear in advance of the actual begin of the event during the event *lead time*, and are still published after the event completion during the *cool-down time*. For unexpected non-recurring events such as natural disasters, event-related documents are published from the start of the event onward, i.e. there is no lead time and the relevant documents appear during the cool-down time of the event. The duration of the lead time as well as the duration of the cool-down time depend on the specific event (see Table 1).

Given an event of user interest and a large-scale broad-scope Web archive, our goal is to generate an interlinked collection of documents relevant to this event. The scope of the target collection is defined in the **Collection Specification**:

Definition 1 (Collection Specification)**.** The Collection Specification defines the topical and the temporal scope of an event-centric collection using:
- Topical Scope:
  - one or more topical reference documents (e.g. from the Web);
  - zero or more representative *keywords*.
- Temporal Scope:
  - time span of the event (including the start and end dates) $T_e = [t_e^s, t_e^e]$;
  - time duration of the lead time (T_l) and the cool-down time (T_r).

The Collection Specification may be extended to include additional scopes, for example domain black and white lists as used by existing crawlers.

Given the Collection Specification, our goal is to create a collection containing the Web documents temporally and topically relevant to this specification. In the following we propose a focused extraction method that prioritizes URLs during the crawling process according to the Collection Specification and generates interlinked event-centric collections.

4 Event-centric Collection Extraction

Our goal is to efficiently extract an event-centric interlinked collection of a manageable size from a large scale Web archive. A naïve approach is to iterate through all documents and check their relevance with respect to the Collection Specification using an automatic method. However, this is computationally expensive and does not scale to Web archives spanning tens or hundreds of terabytes. While a full-text index could reduce the iteration cost, it requires high up-front computational and index storage resources and extensive post-filtering of the many near-identical document versions contained in the Web archive [16]. Furthermore, such an index can only be used to retrieve individual documents, where we want to extract interlinked document collections.

We propose an alternative approach that uses the hypertext characteristics of the archived documents by adapting focused Web crawling. A Web crawler collects documents by recursively following the links from a Web document to

Algorithm 1 Event-centric Collection Extraction

Input: Collection Specification CS, $targetSize$
Output: Document collection c, excluded URLs $missing$

```
q ← priorityQueue(seedUrls(CS)); c ← {}; missing ← {}
while not isEmpty(q) and |c| < targetSize do
    url ← pop(q)
    v ← resolveSnapshots(url, CS) {Find all snapshots of url in c}
    if v = ∅ then
        missing ← missing ∪ {url}
    else
        v_i ← selectSnapshot(CS, v)
        c ← c ∪ {v_i}
        out ← extractOutlinks(v_i) − seenUrls {seenUrls = c ∪ missing}
        insert(q, out, relevance(v_i)) {Insert outlinks into queue according to relevance}
    end if
end while
```

other documents, starting from an initial set of *seed URLs*. A focused Web crawler improves the relevance of the resulting collection by following only links to the documents predicted to be relevant. We therefore extend the Collection Specification to include the seed URLs required for the crawling process:

Definition 2 (Crawl-based Collection Specification)**.** A Crawl-based Collection Specification contains a Collection Specification (Definition 1) and a non-empty set of URLs, which are contained in the archive and refer to relevant documents.

The Crawl-based Collection Specification is created by the user. Semi-automatic approaches include the use of Web search engines to select seed URLs [13].

We adapt the focused crawling algorithm as shown in Algorithm 1 by including steps to resolve snapshots and select the best among them. URLs extracted from collected documents are prioritized in the crawler queue during the focused crawl using the relevance function defined in Section 5.

5 Relevance Estimation

We need to prioritize the URLs during the focused crawl to effectively extract event-centric collections based on a *relevance function*. We use a linear combination of the temporal and topical relevance (TTR) to estimate the relevance of a Web document d with respect to the Collection Specification CS:

$$\mathrm{TTR}(d, CS) = \alpha \times \mathrm{TopicR}(d, CS) + (1 - \alpha) \times \mathrm{TempR}(d, CS), \qquad (1)$$

where $TempR$ and $TopicR$ are the temporal and topical relevance of d to CS, and $\alpha \in [0, 1]$ is the parameter to trade off between the topical and temporal relevance. $\alpha = 1$ results in a standard topically focused crawler, whereas values closer to 0 increase the weight of the temporal dimension. In our setting we consider $TempR$ and $TopicR$ to be equally important, therefore we use $\alpha = 0.5$, but we will in future work investigate the influence of this parameter in detail.

5.1 Temporal Relevance

As described in Section 3, event-related documents are published not only during the event time interval, but also before and after. Consequently, we need to estimate the relevance of a document based on the Collection Specification and a time point associated with the Web document (e.g. the creation, last modification or capture date). We define this *Temporal Relevance Function* as follows:

Definition 3 (Temporal Relevance Function)**.** Given a time point t_d associated with the Web document d and the event time interval $T_e = [t_e^s, t_e^e]$, the function $f(t_d, t_e) \rightarrow [0, 1]$ is a temporal relevance function iff (a) $f(t_d, t_e) = 1 \Rightarrow t_d \in t_e$ and (b) f is monotonically non-decreasing in $(-\infty, t_e^s)$ and monotonically non-increasing in $(t_e^e, +\infty)$.

We assume that in general the relevance of documents decreases rapidly as the distance to the event increases and therefore define a temporal relevance function based on the exponential decay function (similar to [18]):

$$\text{TempR}(t_d, t_e) = \begin{cases} 1 & \text{if } t_e^s \leq t_d \leq t_e^e, \\ e^{-\Delta t/\gamma_l} & \text{if } t_d < t_e^s, \\ e^{-\Delta t/\gamma_r} & \text{if } t_d > t_e^e, \end{cases} \tag{2}$$

where Δt is the time difference between the document time point t_d and the nearest end of the reference time interval T_e, and γ_l and γ_r are *time decay factors*. The time decay factors determine how fast the value of this function decreases by giving the Δt at which the relevance has dropped to 0.5. We use the expected duration of the lead and the cool-down time as the time decay factors γ_l and γ_r. For events with no lead time (e.g. accidents) we set $\gamma_l = 0$.

The document time point can be estimated using the date discussed in the document. This would give the most accurate relevance value, especially for documents that describe the event after some time has passed (e.g. at the one year anniversary), but is computationally expensive and highly heuristic. Therefore we extract the document publication time, which is often explicitly contained in the document metadata or content. If no publication time is available, we use the crawl time as a fallback.

5.2 Topical Relevance Estimation

The topical relevance of Web documents with respect to the Collection Specification is estimated by computing the similarity of the textual content of Web documents to the topical scope of the Collection Specification (similar to [23]).

The topical scope is specified primarily through a set of *reference documents* that describe the event (e.g. as Wikipedia pages or newspaper articles). When these documents have an ambiguous topic or the scope should be narrowed down further, keywords can be provided to clarify the topical intent. Together this allows an intuitive yet powerful topical specification. We represent the topical

scope as a term vector, called the *reference vector*, to enable automatic relevance estimation with respect to the topic. To construct the reference vector we tokenize and stem the text of the reference documents and remove stop words using the language-specific analyzers of Apache Lucene[6]. As previous work has shown bigrams to be effective for crawl focusing [19], we use term unigrams and bigrams. Each term is weighted using its frequency (TF) and its inverse document frequency (IDF). IDF scores are based on the frequencies of the last 25 years of the Google Books NGram datasets[7].

The weights of terms explicitly given as Collection Specification keywords are boosted. This helps to shift the reference vector towards the expected interpretation. To perform boosting, we check the overlap of each term with the user-defined keywords, as terms (in the case of bigrams) can contain multiple tokens. Based on whether there is a full or partial overlap, we assign a *term weight* tw_t to the term t in the document vector. In our evaluation, we experimentally set the values for full, partial and no overlap to 2, 1.5 and 1, respectively.

Finally, the topical relevance of a document is the cosine similarity between the reference vector and a document vector computed using the same method.

6 Web Archive and Platform

Our Web Archive contains all Web pages from the `.de` top-level domain as captured by the Internet Archive until 2013. In this paper we only consider HTML documents with a HTTP status code of 200. This archive has a size of about 30 TB and contains 4.05 billion captures of 1 billion URLs, covering a time period from December 1994 to September 2013.

We manually defined 28 events to be extracted from the Web archive, focusing on events that are likely to be represented in the archive: The selected events fall within the time period of the archive and have a strong connection to Germany, either because the event happened in Germany or was in the focus of public attention. We balanced singular events like the Fukushima nuclear accident and recurring events like federal elections. To create the Collection Specification for each event we selected one or more pages from the German Wikipedia that provide the topical scope of the event. We also defined a start and end date, as well as an estimate for the duration of the event lead and cool-down time. The outgoing links of the Wikipedia pages were extracted and used as seed URLs.

All experiments were conducted on a Hadoop cluster. This cluster has 25 worker and 2 master nodes with in total 296 CPU cores. The worker nodes provide in total 1.37 TB of RAM and 1 PB of hard disk capacity. All data is stored in the standard ARC/WARC formats and available to all worker nodes.

6.1 Crawler Implementation

As mentioned in Section 4, the architecture of the archive crawler can be simpler than that of a standard Web crawler because it can access the data of the Web

[6] `http://lucene.apache.org/core/`

[7] Code available at: `https://github.com/gerhardgossen/dictionary-creator/`

archive locally. As our data is stored as WARC files in a Hadoop filesystem, we implemented the crawler as a multi-thread process running on Hadoop YARN.

WARC files are unordered collections of documents, therefore a lookup table is necessary to find the location of the document snapshots for a given URL. By using Apache HBase for this table we can look up URLs in 1-5 milliseconds. While typically CDX files are used as a lookup method for WARC files, our preliminary experiments showed that this method is considerably faster.

The crawler queue is stored in a file-based queue based on the Mercator architecture [15], which offers prioritisation of URLs and is fast enough for our purposes. Each retrieved document is analysed according to the relevance function described in Section 5. The URLs of all outgoing links of that document are inserted into the crawler queue according to the calculated relevance score.

As the Web archive covers a long time period, many documents have been crawled multiple times. To choose among the available versions, we observe that later versions typically have the same content but may have changes in e.g. navigation menus and thus do not represent the document in its original form. Therefore we use the following heuristic: If multiple versions are available that were crawled during the event timespan, we pick the earliest. Otherwise, we use the version that was crawled closest to the event timespan. Future work will investigate further methods to select the most relevant version(s).

7 Evaluation

The goal of the evaluation is to assess the precision of the proposed extraction method in light of different event types and to better understand the influence of this method on the quality of the resulting event-centric collections. We compare our combined relevance function with two baselines that use state-of-the-art relevance functions, each taking only one relevance dimension into account, topical (C-F, cf. [23]) or temporal (T-F, cf. [24]). We also use an unfocused crawl that does not use any relevance estimates as an additional baseline.

7.1 Extraction Evaluation

Our focused crawling approach allows us to adjust the effort invested into the extraction by changing the number of documents processed. By increasing this number to the size of the archive we could clearly guarantee that this method finds all the relevant documents, as long as they are reachable through links. However, the proposed approach should be able to extract most of the relevant documents early on, so that the extraction can be stopped when not sufficiently many relevant documents are discovered anymore or when the user is satisfied with the collection. We therefore look at the accumulated relevance (i.e. the sum of the relevance values of the extracted documents) of the collected results as a function of crawl runtime. extraction process. Additionally, we look at the number of documents that the crawler attempts to capture but are missing from the archive.

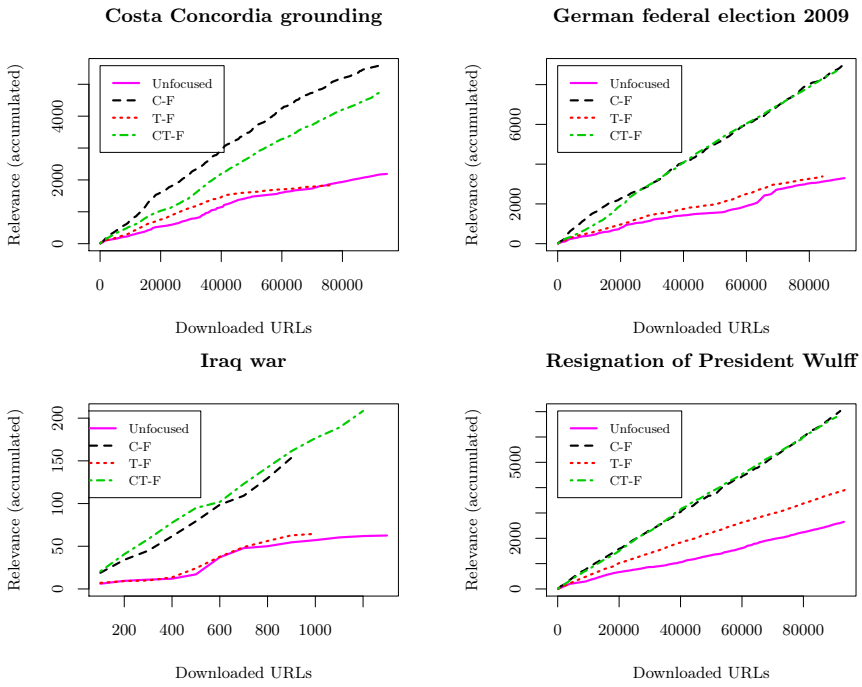

Fig. 1. Accumulated relevance of different event collections.

The relevance of the extracted documents is computed with the C-F relevance function. This is possible because we estimate the relevance of a document during the crawl using the content of a linking document and evaluate using the content of the actual document. A small annotation experiment (omitted for space) showed that this relevance measure correlates with the actual relevance.

For each of the 28 events we started a crawl using each of the configurations described above. Each crawl ran until it had retrieved 100,000 documents or until the crawler queue was empty. Fig. 1 shows the accumulated relevance of document collections for selected events in relation to the number of documents crawled. This function should ideally start with a strong incline, meaning that the crawler fetches many relevant documents early on, flattening into a plateau when no relevant documents are available anymore. We see that for all topics the C-F and CT-F functions outperform the T-F function and the unfocused baseline both in terms of average relevance of documents retrieved at any given point and total relevance. The C-F function often performs slightly better than the CT-F function, although closer analysis shows that the differences between both functions often result from discovering some highly relevant hosts earlier.

The relevance focused strategies manage to uncover more potentially relevant URLs even if they are not contained in the locally available Web archive. This is shown by the number of URLs that each focusing method considers (see Table 2), where we see an increase in discovered URLs for these methods. Based on this

Table 2. URLs considered for each event crawl for different relevance strategies.

Topic	CT-F	Unfocused	Ratio
Costa Concordia grounding	239,628	142,851	1.67
German federal election 2009	283,311	161,934	1.74
Iraq War	1,862	2,192	0.84
Pope Election 2013	2,057	1,624	1.26
Stuttgart 21 protests	2,070	1,513	1.36
Resignation of President Wulff	213,039	149,706	1.42

Table 3. Effect of temporal scope and keyword parameters. Each row shows improvement ratio of the accumulated relevance for a topic with event-specific time parameters (left) or keywords (right). The last line contains the average improvement over all topics. All values are statistically significant at $p = 0.01$.

Event	**Time**		**Keywords**		**Event**	**Time**		**Keywords**	
	T-F	CT-F	C-F	CT-F		T-F	CT-F	C-F	CT-F
Book by Thilo Sarrazin	0.98	0.99	1.28	1.07	Iraq war	0.92	1.19	1.05	1.13
Eruption of Eyjafjallajökull	0.99	1.20	0.83	0.88	Launch of LHC	1.09	0.72	1.21	0.99
European Stability Mechanism	1.16	4.07	1.02	1.04	Costa Concordia grounding	1.14	1.49	0.92	0.98
European floods 2013	1.12	1.12	1.39	1.49	Loveparade disaster	0.84	1.25	0.81	0.97
Eurovision Song Contest 2010	1.00	1.73	1.06	0.68	NSU process	1.01	1.24	1.05	1.05
Football World Cup 2006	0.58	1.27	1.23	1.10	Olympia 2004	0.94	1.03	1.20	1.34
Football World Cup 2010	1.59	1.09	1.11	1.10	Olympia 2008	1.27	1.48	1.39	1.50
Fukushima nuclear disaster	1.17	1.73	1.03	1.02	Olympia 2012	1.18	1.11	1.16	1.12
German federal election 2002	1.21	1.48	1.35	1.02	Olympia 2010	1.02	1.37	1.24	1.65
German federal election 2005	1.33	1.41	1.14	0.89	Pope Election 2005	1.17	1.08	1.10	1.09
German federal election 2009	1.27	1.84	1.03	0.96	Pope Election 2013	1.07	1.50	0.99	0.95
German federal election 2013	1.12	2.17	0.84	0.92	Resig. of Pres. Wulff	1.03	1.05	1.00	1.03
Guttenberg plagiarism affair	0.96	1.01	1.24	1.19	Snowden leaks	1.46	1.43	1.18	1.19
average						1.10	1.44	1.10	1.08

result, the development of methods for cross-archive collection extraction is an interesting direction for future research.

7.2 Effect of the Temporal Scope Parameters

In the Collection Specification we require that the user specifies *lead* and *cool-down times* for the event (cf. Section 5.1) to adapt the temporal relevance function to different event types. We crawled each event using a exponential decay function with a fixed decay and compared it to the crawl using the specified lead and cool-down times. Table 3 (left columns) shows the relevance improvement of the time-sensitive relevance functions over the corresponding baseline. We see that the event-specific parameters cause an improvement for most of the events. On average this improvement is moderate, but statistically significant.

7.3 Effect of Keywords in the Specification

We use the keywords in the Collection Specification to clarify the topical intent (cf. Section 5.2). To measure the impact, we crawled using the same reference

documents with and without keywords to describe the topical scope. Table 3 (right columns) shows the relevance improvement of the T-F and CT-F relevance functions compared to the corresponding baseline. We see that the addition of keywords leads on average to a statistically significant improvement. Some events such as the floods in Europe during 2013 can be better focused using keywords, whereas for other events adding keywords leads to a small loss in effectiveness. Further research is needed to better understand the influence of keywords.

8 Conclusions and Outlook

In this work we presented a novel method to create interlinked event-centric collections from large-scale Web archives. The key of this method is to adapt focused Web crawling to previously collected Web archives and to select documents by iteratively following links from relevant documents. We proposed relevance estimation functions that take the temporal and topical aspects of the documents into account and evaluated them as part of the focused extraction process. Specifically, we demonstrated that the relevance function CT-F can improve on topical content selection methods by taking temporal information into account. This holds especially for events that occur repeatedly in similar form, such as Olympic games or elections, where the different instances are hard to distinguish using only topical information. We showed that our re-crawling method can retrieve event-centric collections from large-scale Web archives, especially using the CT-F relevance function, and discussed how the method deals with the challenges inherent to Web archives.

Our method presents a first step towards the extraction of event-centric collections. Further research is needed to understand the influence of extraction methods, relevance functions and parameters in regard to different events, time periods and Web archives. For Web archives that have full-text indexes, methods based on full-text search should be investigated. Furthermore, cross-archive collection extraction is an interesting direction for future research. We therefore provide our source code and evaluation data to encourage similar efforts [8].

Acknowledgments

This work was partially funded by the ERC under ALEXANDRIA (ERC 339233), H2020 under SoBigData (RIA 654024) and BMBF under Data4UrbanMobility (02K15A040).

References

1. Aggarwal, C., Al-Garawi, F., Yu, P.S.: Intelligent crawling on the world wide web with arbitrary predicates. In: World Wide Web Conference. pp. 96–105 (2001)
2. Berberich, K., Bedathur, S.: Temporal diversification of search results. In: Workshop on Time-aware Information Access (TAIA 2013) (2013)

[8] `https://github.com/gerhardgossen/archive-recrawling`

3. Bergmark, D., Lagoze, C., Sbityakov, A.: Focused crawls, tunneling, and digital libraries. In: ECDL '02 (2002)
4. Chakrabarti, S., van den Berg, M., Dom, B.: Focused crawling: a new approach to topic-specific web resource discovery. Computer Networks 31(11-16) (1999)
5. Costa, M., Couto, F., Silva, M.: Learning temporal-dependent ranking models. In: SIGIR '14 (2014)
6. Costa, M., Gomes, D., Silva, M.J.: The evolution of web archiving. IJDL (2016)
7. Diligenti, M., Coetzee, F., Lawrence, S., Giles, C.L., Gori, M.: Focused crawling using context graphs. In: VLDB (2000)
8. Dong, A., Chang, Y., Zheng, Z., Mishne, G., Bai, J., Zhang, R., Buchner, K., Liao, C., Diaz, F.: Towards recency ranking in web search. In: WSDM'10 (2010)
9. Dong, H., Hussain, F.K.: SOF: a semi-supervised ontology-learning-based focused crawler. Concurrency and Computation: Practice and Experience 25(12) (2013)
10. Ehrig, M., Maedche, A.: Ontology-focused crawling of web documents. In: ACM SAC (2003)
11. Farag, M.M.G., Lee, S., Fox, E.A.: Focused crawler for events. IJDL (2017)
12. Gossen, G., Demidova, E., Risse, T.: iCrawl: Improving the freshness of web collections by integrating social web and focused web crawling. In: JCDL '15 (2015)
13. Gossen, G., Demidova, E., Risse, T.: The iCrawl Wizard – supporting interactive focused crawl specification. In: ECIR'15 (2015)
14. Gossen, G., Demidova, E., Risse, T.: Analyzing web archives through topic and event focused sub-collections. In: WebSci '16. pp. 291–295 (May 2016)
15. Heydon, A., Najork, M.: Mercator: A scalable, extensible web crawler. World Wide Web 2(4), 219–229 (1999)
16. Jackson, A., Lin, J., Milligan, I., Ruest, N.: Desiderata for exploratory search interfaces to web archives in support of scholarly activities. In: JCDL'16 (2016)
17. Jiang, J., Song, X., Yu, N., Lin, C.Y.: Focus: Learning to crawl web forums. IEEE TKDE 25(6) (2013)
18. Kanhabua, N., Nørvåg, K.: A comparison of time-aware ranking methods. In: SIGIR '11 (2011)
19. Laranjeira, B., Moreira, V., Villavicencio, A., Ramisch, C., Finatto, M.J.: Comparing the quality of focused crawlers and of the translation resources obtained from them. In: LREC '14 (2014)
20. Mohr, G., Kimpton, M., Stack, M., Ranitovic, I.: Introduction to Heritrix, an archival quality web crawler. In: 4th International Web Archiving Workshop (2004)
21. Nguyen, T.N., Kanhabua, N., Niederée, C., Zhu, X.: A time-aware random walk model for finding important documents in web archives. In: SIGIR '15 (2015)
22. Pant, G., Srinivasan, P.: Learning to crawl: Comparing classification schemes. ACM Transactions on Information Systems 23(4) (2005)
23. Pant, G., Srinivasan, P., Menczer, F.: Crawling the web. In: Web Dynamics (2004)
24. Pereira, P., Macedo, J., Craveiro, O., Madeira, H.: Time-aware focused web crawling. In: ECIR'14 (2014)
25. Qin, J., Zhou, Y., Chau, M.: Building domain-specific web collections for scientific digital libraries. In: JCDL'04 (2004)
26. Risse, T., Demidova, E., Gossen, G.: What do you want to collect from the web? In: Proc. of the Building Web Observatories Workshop (BWOW) 2014 (2014)