Int J Digit Libr
DOI 10.1007/s00799-017-0215-9

Quantifying retrieval bias in Web archive search

Thaer Samar[1] · **Myriam C. Traub**[1] · **Jacco van Ossenbruggen**[1,3] · **Lynda Hardman**[1,4] · **Arjen P. de Vries**[2]

Received: 1 March 2017 / Revised: 7 March 2017 / Accepted: 8 March 2017

Abstract A Web archive usually contains multiple versions of documents crawled from the Web at different points in time. One possible way for users to access a Web archive is through full-text search systems. However, previous studies have shown that these systems can induce a bias, known as the *retrievability bias*, on the accessibility of documents in community-collected collections (such as TREC collections). This bias can be measured by analyzing the distribution of the *retrievability scores* for each document in a collection, quantifying the likelihood of a document's retrieval. We investigate the suitability of retrievability scores in retrieval systems that consider every version of a document in a Web archive as an independent document. We show that the retrievability of documents can vary for different versions of the same document and that retrieval systems induce biases to different extents. We quantify this bias for a retrieval system which is adapted to handle multiple versions of the same document. The retrieval system indexes each version of a document independently, and we refine the search results using two techniques to aggregate similar versions. The first approach is to collapse similar versions of a document based on content similarity. The second approach is to collapse all versions of the same document based on their URLs. In both cases, we found that the degree of bias is related to the aggregation level of versions of the same document. Finally, we study the effect of bias across time using the retrievability measure. Specifically, we investigate whether the number of documents crawled in a particular year correlates with the number of documents in the search results from that year. Assuming queries are not inherently temporal in nature, the analysis is based on the timestamps of documents in the search results returned using the retrieval model for all queries. The results show a relation between the number of documents per year and the number of documents retrieved by the retrieval system from that year. We further investigated the relation between the queries' timestamps and the documents' timestamps. First, we split the queries into different time frames using a 1-year granularity. Then, we issued the queries against the retrieval system. The results show that temporal queries indeed retrieve more documents from the assumed time frame. Thus, the documents from the same time frame were preferred by the retrieval system over documents from other time frames.

Keywords Web archive · Retrieval bias · Evaluation

1 Introduction

Indexing and retrieving documents from a Web archive can be challenging. Web archive collections are different from conventional static Web collections. The main reasons are the continuously increasing size of Web archives and the existence of multiple versions of the same document collected at different moments in time. The different versions may appear multiple times in search results and thereby render other documents inaccessible for a user. Despite these challenges, Web archive initiatives make an effort to make their collections better accessible. For example, Gomes et al.

✉ Arjen P. de Vries
a.devries@cs.ru.nl

Thaer Samar
thsammar@gmail.com

[1] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

[2] Radboud University, Nijmegen, The Netherlands

[3] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

[4] Universiteit Utrecht, Utrecht, The Netherlands

Published online: 18 April 2017

[27] conducted a survey in 2010 on 42 Web archive initiatives around the world (26 countries). They found that 89% of the initiatives support access to the Web archive by a given URL, 79% support searching metadata, and 67% provide *full-text* search over their archives. The same survey was conducted again in 2014 in order to observe the change in Web archiving since 2010 [18]. They noticed an increase in the number of initiatives (68) and the number of countries involved in Web archiving (33 countries). However, in terms of access methods, the results of 2014 are the same as those for 2010.

Previous studies showed that applying existing Information Retrieval (IR) models on Web archives leads to unsatisfactory results [17,21]. Measuring the effectiveness of IR systems can be done using test collections. A test collection consists of a set of topics (queries), a document collection, and a set of relevance assessments. Costa and Silva [21] extended this approach by taking the characteristics of Web archives into account. Their approach includes the design of a test collection and constructing topics from the users' query log of a functioning Web archive search system. Getting relevance judgments, however, is a costly process. An additional complication is the dependency on query logs, as they are seldom available.

To complement standard methods of IR evaluation, that focus on the assessment of efficiency and effectiveness of IR systems, Azzopardi et al. introduced *retrievability* as a measure for potential bias in the access of documents in a collection [7]. The retrievability score of a document counts how often the document is retrieved when a large representative set of queries is issued to the retrieval system. The overall bias in the scores among all documents in the collection induced by a retrieval system can be quantified using measures such as the Lorenz Curve [26] and the Gini Coefficient [26]. While the Lorenz Curve can be used to visualize the bias, the Gini Coefficient can be used to quantify the extent of bias for different experimental conditions.

We follow an approach similar to [7] to study how retrievability can be used to quantify retrieval bias induced by different retrieval systems on a subset of the Dutch Web archive collection from the National Library of The Netherlands[1] (*KB*).

Our main goal is to investigate how to use retrievability to evaluate a Web archive retrieval system, and how the number of document versions and the method of aggregation of crawls influence the retrieval bias in the Web archive.

Specifically, we address the following research questions:

RQ1 *Is access to the Web archive collection influenced by a retrievability bias? Can we evaluate and compare retrieval systems on the Web archive collection using the retrievability measure to quantify their retrieval bias?*

We follow the approach of [7] to quantify the overall bias imposed by different retrieval systems using the Gini Coefficient and the Lorenz Curve constructed using retrievability scores of documents in the collection.

RQ2 *How does the number of versions of documents in the Web archive collection influence the retrievability bias of a retrieval system?*

The number of versions per document in the archive varies, for example, because documents have been crawled with different frequencies or because they were added to the crawler's seed list at different points in time. We show how multiple versions impact the retrieval bias when the granularity of retrieval in the search results is the document's version (each version of a document is considered an independent document). We compute the retrievability score of a document by accumulating the retrievability score of its versions: a document with more versions is assigned a higher retrievability score. We show the change in bias when the multiple versions are handled by the retrieval system using two approaches to collapse documents' versions: first, based on their content similarity; second, based on their URLs.

RQ3 *Does a retrieval system favor specific subsets of the collection?*

The Web archive collection of the *KB* consists of snapshots of websites from different points in time spanning 4 years. Therefore, we investigate what subset of the archive is most affected by retrieval bias.
The remainder of the paper is organized as follows. After discussing related work (Sect. 2), we describe our approach to answer the research questions introduced in this section (Sect. 3). We discuss the experimental setup in detail in (Sect. 4) and answer research questions *RQ*1-3 in Sects. 5, 6, 7, respectively. Finally, we discuss conclusions drawn from our findings (Sect. 8).

2 Related work

Understanding the information needs of Web archive users is an important step toward developing good access methods for Web archives. Several studies showed that *full-text* search is preferred [19,20,27,41]. This shift from single URL search to search interfaces was described as a turning point in the history of Web archives [13].

Research in *temporal IR* aims to exploit temporal information in documents and queries for better query understanding

[1] www.kb.nl.

and time-based ranking [1,16,32]. Costa and Silva [21] created a temporal test collection from the Portuguese Web Archive [28], to enable evaluation of temporal methods in IR. A test collection consists of queries (topics), documents, and the judgments by users of their relevance to the queries. When a new system is built then its effectiveness can be measured based on the test collection using evaluation metrics such as precision (for example *P@10*). The collection developed by Costa and Silva consists of crawls in the period from 1996 to 2009. The queries (topics) were selected from query logs, and the documents retrieved by the retrieval system were manually judged. Their method extends the Cranfield paradigm with consideration of the temporal aspect of Web archive collections. Other studies used crowdsourcing to collect relevance judgments. For example, Berberich et al. [14] used Amazon Mechanical Turk to collect queries and relevance assessments.

Retrievability was introduced to measure how likely a document is to be retrieved given an IR system [5–7]. Computing the retrievability scores requires the availability of a large query set, but without the need for relevance judgments. Queries can be simulated by drawing them from the content of documents in the collection. The retrievability score of a document $r(d)$ gives an indication of how retrievable the document is compared to other documents in the collection. It is computed by accumulating the number of times this document appears in the ranked list provided for all queries, at a given cutoff rank. In order to quantify the retrievability bias across all documents in the collection, the Lorenz Curve [26] is used to visualize the bias and the Gini Coefficient [26] is used to summarize the bias. In economics, the Lorenz Curve is used to visualize the distribution of wealth or income of a population. If the wealth or income is equally distributed in the population, the accumulative distribution is a diagonal line (called the line of equality). The larger the inequality is within a population, the more the curve deviates from the equality line. The Gini Coefficient summarizes the overall inequality into a value which ranges from zero (perfect equality) to one (perfect inequality). The Gini Coefficient quantifies the retrievability inequality among documents. In the context of retrievability, the population corresponds to the document collection and wealth corresponds to the retrievability scores.

Retrievability has been used to compare different retrieval models based on the bias they impose on a given collection, and to study whether the retrieval system favors documents with particular features. For example, the system might favor long documents over shorter documents. In the following, we discuss a few studies that used retrievability. Retrievability was applied in the patent search domain [8,11], which is recall-oriented, to quantify the retrieval bias of retrieval systems on the patent collection. The correlation between retrievability and the query set was considered in several studies. Based on a limited set of queries, the correlation between retrievability score and query relevance to the document[2] was analyzed [9]. Their experimental results showed that 90% of highly retrievable documents when all queries were considered are not highly retrievable considering only their relevant queries. The influence of query characteristics on retrieval bias was explored in [12]. They showed that different query characteristics increase or decrease the retrieval bias differently. Query expansion was used to improve document's retrievability [10].

Other studies investigated the relation between a system's retrieval bias and its effectiveness. For example, Azzopardi et al. [2] showed that a positive relation exists between effectiveness and retrievability. Measuring effectiveness using precision at 10 ($P@10$) & Mean Average Precision (MAP), the results showed that as the effectiveness increases, the retrievability bias tends to decrease. This relationship between retrievability and effectiveness has been used to tune systems [44]. Bashir and Rauber [10] investigated the impact of query expansion on the retrievability bias. They showed that standard query expansion methods caused an increase in effectiveness and retrieval bias. They explained the increase in retrieval bias due to the assumption of query expansion methods that the top-ranked documents are relevant. However, some documents in the top-ranked results might be noise. Therefore, in order to decrease the retrieval bias, they proposed a query expansion approach based on document clustering, and they showed that their approach reduces the bias.

3 Approach

We explore how we can use retrievability to assess the retrieval bias of retrieval systems providing access to 4 years of the Dutch Web archive. In order to investigate our first research question, *RQ1*, we use three well-known IR models and two large query sets. For every model and query set, we compute the retrievability score ($r(d)$) for document versions at different rank cutoffs c. Parameter c represents the willingness of the user to explore a certain number of documents in the search results; therefore, it is independent from the retrieval model. In our study, we experiment with $c = 10$, 20, 30, 40, 50, 100, and 1000. Users are known to rarely evaluate more than the first 10 search results; however, we also consider high values for c to find out whether the inequality bias would still exist if the users were willing to explore higher numbers of results. In order to allow the comparison of the retrieval models in terms of retrieval bias they impose on the documents, we need a measure to quantify the overall

[2] The relevance of the document to each query in a small sample was assessed by experts.

bias given a collection, a query set, and a retrieval system. We use the Gini Coefficient to summarize the retrieval bias, and the Lorenz Curve to visualize the retrieval bias, following [7].

A certain fraction of documents is *not-retrieved* by any of the retrieval models. This fraction is especially high for smaller c's and has a strong influence on the overall bias measured by the Gini Coefficient. Therefore, we compute two variants of the Gini Coefficient. In the first variant, all documents in the collection are included; if a document is not retrieved by the model, its retrievability score is zero ($r(d) = 0$). Here, the number of documents is the same for all models at all c's (number of retrieved documents plus number of not-retrieved documents = whole collection). In the second variant, only documents that are retrieved using at least one of the three retrieval models at a given c are considered. We do this by creating a union set of unique documents retrieved using at least one of the three models at the given c (*3Models_union_c*) for each query set. If a document was retrieved using model A, but not with model B, then the retrievability score of that document given model B is assigned a value of 0 ($r_B(d) = 0$). The number of documents will be the same for all models at the same c (num. retrieved plus num. not-retrieved = *3Models_union_c*). Therefore, this can still be considered to provide a fair comparison across the retrieval models for a given c. Using the second variant will reduce the impact of a high fraction of documents with $r(d) = 0$. A model that does not retrieve a large number of documents that were retrieved using other models will get a higher Gini Coefficient, that is, it is considered to be more biased.

In order to understand the relation between the retrievability scores and the ability to find a document in the collection, we use a known-item-search setup based on the approach proposed in [3,4].

We quantify the impact of multiple versions of the same document on the inequality of retrieval bias, *RQ2*. First, we investigate the retrieval of all versions of a document. At indexing and retrieval time, we consider the document's version as an independent document. In order to check how that affects the document's retrievability, we compute the retrievability of a document by aggregating the retrievability scores of its versions retrieved at a given c, and thus the overall bias imposed by the model. Second, we collapse similar versions of the same document and again compute the retrievability score and the overall bias. Third, to explore the impact of the number of versions on the bias, we linearly combine the scores given by the models with a prior based on the number of versions. This allows us to measure retrieval bias at the granularity of the document, instead of a specific version.

Finally, we address our last research question, *RQ3*. Our Web archive collection is an accumulation of several crawls over time. We are interested in whether the bias imposed by a given retrieval system, among subsets based on the time of crawling, correlates with the number of crawled documents in that year. To explore this research question, we focus on the documents retrieved using the *BM25* model; as we show in the results, it provides the least bias. Using the timestamps of the crawling time associated with documents, we split the search results for *BM25* into four subsets at different c's and then measure the retrieval bias per subset.

4 Experimental setup

In Sect. 4.1, we describe the components used to measure retrievability on the Web archive collection. In Sect. 4.2, we describe the known-item search setting to investigate the relation between retrievability score of a document and the difficulty level of finding that document.

4.1 Retrievability experimental setup

First, we introduce the Dutch Web archive collection (Sect. 4.1.1). Then, we describe how we preprocessed and indexed the collection (Sect. 4.1.2). After that, we discuss how we designed the query sets that are used to retrieve documents from the collection (Sect. 4.1.3). Finally, we discuss how to measure retrievability scores and how to quantify the overall bias imposed by a given retrieval model (Sect. 4.1.4).

4.1.1 Data set

In their Web archive, the *KB* preserves a growing seed set of currently more than 10,000 websites [40]. For our research, the KB provided us with a subset of the Dutch Web archive that has been harvested between February 2009 and December 2012, consisting of 76,828 Archive (*ARC*)[3] files. Each ARC file contains multiple archived records (content plus the response header), which yields a total of 148M documents. Table 1 shows the total number of archived objects, raw count and the percentage of *text/html*. We refer to *text/html* content-type objects as documents. These documents form our collection D on which we focus our analysis. Every crawled document has its own URL and the timestamp of the crawling time in addition to its content on the Web at the time of the crawl. Every document d may have multiple versions crawled at different points in time t_i,

$$d := \left\{d_v^{t_1}, d_v^{t_2}, \ldots, d_v^{t_n}\right\}$$

where $d_v^{t_1}$ is the *document's version* crawled at time t_1. The mean value of number of versions (total number of versions over the number of unique documents based on URLs)

[3] http://archive.org/web/researcher/ArcFileFormat.php.

Table 1 Summary of the archived objects over the years, with more details on documents of *text/html* content-type

Year	Archived objects all types	Text/html documents			
		All versions (mementos)	%	Original URLs	Mean (#versions)
2009	17,014,067	12,232,831	71.9	9,764,370	1.25
2010	38,157,308	22,596,291	59.2	17,093,870	1.32
2011	53,604,464	30,275,150	56.5	19,491,258	1.55
2012	38,865,673	19,464,431	50.1	13,191,771	1.48
All	147,641,512	84,568,703	57.3	47,836,163	1.77

The mean value of number of versions was computed by dividing the total number of document versions crawled per year over the unique number of documents (URLs). The number of original URLs for *All* years is the number of unique URLs in the 4 years

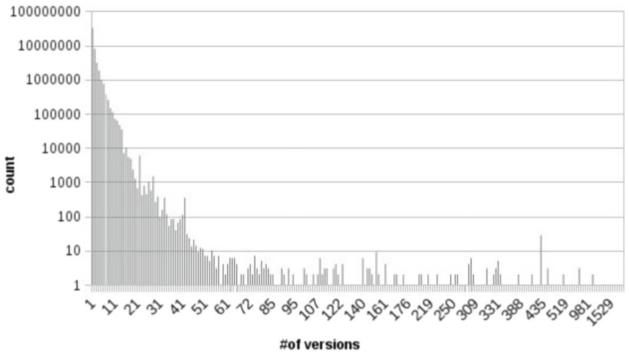


Fig. 1 Distribution of number of versions of documents in the Dutch Web archive collection in log scale representation

increases over the years, as more crawls have been added to the archive (see Table 1). The distribution of the number of versions per document is skewed (see Fig. 1 in a log scale).

4.1.2 Preprocessing and indexing

Preprocessing consists of removing HTML tags, tokenization, removing stopwords, removing terms of length less than 3 characters, removing numbers with fewer than 4 digits, and stemming. For every document's version $d_v^{t_i}$, we keep the following data:

$$d_v^{t_i} := \{\text{URL, docId, crawl-date, pre-processed-content}\}$$

where the docId is a unique identifier defining the document's version, while the URL is the same for all versions of the same document. We used the Lemur toolkit[4] to index our collection. The documents in our collection are in Dutch, but unfortunately, a Dutch stemmer is not available in the Lemur toolkit. Therefore, we applied stemming in the preprocessing stage[5] and switch off stopword removal and stemming at indexing time (as these have already been applied in the preprocessing stage). The index granularity is the document's version $d_v^{t_i}$. For indexing and retrieval, we used the same IR systems as [7], motivated by their widespread application in IR [37]: *BM25*, *TF*IDF* and *LM1000* (Language Modeling with Bayes Smoothing, $\mu = 1000$).

4.1.3 Query set

In order to compute the retrievability score of all documents in the collection, we need a set of queries to run against a given retrieval system. Ideally, we would use queries collected from users searching the collection. Unfortunately, such a query log is not available for the Web archive. However, there are reasonable alternatives for generating the query set. First, we follow the approach used in [7] by simulating the queries from the content of the documents in the collection. Second, we use the hyperlink's anchor text in the Web archive. One of the defining properties of the Internet is its hyperlink-based structure. The structure of the Web graph is defined by its hyperlinks which consist of a source URL, a destination URL, and an anchor text describing the destination. The hyperlink structure is a rich source of information about the content of a Web collection which has been widely used, especially in the context of Web retrieval, including the *PageRank* algorithm for ranking Web documents [39], and Kleinberg's approach to infer hubs and authorities [34]. Empirical studies have shown that anchor texts exhibit characteristics similar to both user queries and document titles [24]. Language models generated from document titles also can be used as an approximation of a user query language model [30]. Anchor text has been widely used in the IR field to improve search effectiveness [22,23,25,31,35,36,38]. In summary, anchor texts are related to real queries, and target documents' titles. In addition to this, anchor text is available not only for pages in the archive, but also for pages that have not been archived when there are pointers to them from pages in the Web archive [29,33,42].

Simulated query sets The first choice for generating a large set of queries is to draw them from the textual content of documents in the collection following [7]. Their approach exploits the idea behind query based sampling [15], a method that summarizes the content of a database in a non-cooperative distributed search setting starting with a set of keywords. From the preprocessed documents, as described in Sect. 4.1.2, we generate queries of one or two terms. The single-term query set was constructed by taking the most frequent 2 million terms in the collection. The frequencies of the single-term queries range from 5 to 204,517,438. The

[4] http://www.lemurproject.org/.

[5] https://lucene.apache.org/core/4_3_0/analyzers-common/org/apache/lucene/analysis/nl/DutchAnalyzer.html.

Table 2 Summary of the query sets

Query set	# of queries	Mean query length	# of terms
Q_s	4,000,000	1.5	2,000,000
Q_a	1,763,668	2.4	755,589

bi-term query set was constructed by generating all possible two consecutively occurring terms (*bigrams*) from the content of the preprocessed documents. Then, we selected the first 2 million bigrams after ranking them based on number of occurrences. The frequencies of the bi-term queries range from 20 to 35,490,632. The single-term and bi-term queries constitute query set Q_s (4 million queries).

Anchor text query set The second set of queries consists of anchor text constructed from links which we extract from the collection. A link consists of the source URL (the URL of the page where the link was placed), target URL (the URL of the page that the link points to), and the anchor text of the link (a short text describing the target page). To extract the links from the archive, we process all archived Web objects contained in the archive's ARC files. During the processing, JSoup[6] was used to extract links. For each found anchor link, we keep the source URL, the target URL, and the anchor text. We extract the crawl date from a document's metadata, and combine the date with the link information. More precisely, we keep:

$$\langle sourceURL, targetURL, anchorText, crawlDate\rangle$$

We only use the anchor text from external links, where the domain name of the source URL is different from that of the target URL (an inter-domain link). Different seeds are harvested at different frequencies: while most sites are harvested only once a year, some sites are crawled more frequently. Therefore, we deduplicate the links based on their values for source, target, anchor text, and the year of the crawl date. We aggregate the link entries by anchor text and sort them based on their frequency (number of times used to point to the target). Finally, we apply stopword removal and stemming; we refer to this query set as Q_a.

Summary of query sets Table 2 provides the total number of queries, average query length based on the number of terms used per query, and the total number of terms used in each query set (vocabulary of each query set). The number of terms in the vocabulary of the Q_s query set is high. Recall that the simulated queries were extracted from the content of the documents after preprocessing. The terms that were excluded are the Dutch stopwords, terms of length less than 3 characters, and numbers of less than 4 digits. Terms that pass these filters are included, such as numbers, for example dates, telephone numbers, and terms in different languages. After calculating the frequency of terms in the Q_s query set (i.e., the number of queries using each term), we found that a high percentage (45%) of terms were used by one query.

Table 3 Percentage of overlap between the vocabulary of the query sets at different cutoff levels after sorting terms in descending order

Top-c	% of overlap
Top-10k	62.1
Top-50k	57.1
Top-100k	49.8
Top-200k	43.0
Top-300k	34.3
Top-500k	27.3

Table 4 Query length distribution of queries in the Q_a query set

Query length	Number of queries	Percentage
1	397,892	22.6
2	578,819	32.8
3	444,093	25.2
4	247,381	14.0
5	84,463	4.8
6	10,993	0.6

We found that there are 357,258 terms in the overlap between the vocabulary of the two query sets, which is 47.3% of terms in the Q_a vocabulary, and 18.0% of the Q_s vocabulary. To get insights whether the terms in the overlap are the most or least frequent terms, we sorted the vocabulary terms of each query set in descending order based on their frequency; a term frequency is the number of queries using that term. Then, we computed the percentage of overlap at different rank cutoff levels. The percentage of overlap was decreasing by increasing the cutoff of the top frequent terms (see Table 3). In terms of query length, the mean query length (number of terms) of the Q_s query set is 1.5 terms; half of the queries are single-term, and the other half are bi-term queries. The mean query length is 2.4 terms for the Q_a query set. 22.6% of the queries are single-term queries, 32.8% are bi-terms queries, and 25.2% are three-terms queries (see Table 4).

4.1.4 Retrievability assessment

For each of the three IR models discussed above, we issue queries in the query set Q, where $\{Q := Q_s, Q := Q_a\}$. For each $q \in Q$, we collect a ranked list of 1000 documents. Each document in the ranked list has an associated score representing its estimated relevance to the query, and a number representing its position in the ranked list for the retrieval model. The retrievability $r(d)$ of a document d with respect to an IR model given a query set Q is defined as follows (see also [7]):

[6] http://jsoup.org/.

$$r(d) = \sum_{q \in Q} o_q \cdot f(k_{dq}, \{c, g\}) \quad (1)$$

where q is a query from a query set Q, k_{dq} is the rank at which document d is retrieved for q, and $f(k_{dq}, \{c, g\})$ is the access function which indicates how retrievable is d for given q at rank cutoff c. The parameter c represents the effort that the user makes to explore more documents from the provided ranked list. In other words, $f(k_{dq}, \{c, g\}) = 1$ if d is retrieved for q in the given c, otherwise $f(k_{dq}, \{c, g\}) = 0$. For each query set and retrieval model, we compute the retrievability score for all documents in the collection using different $c \in \{10, 20, 30, 40, 50, 100, 1000\}$. Based on Eq. 1, the more queries retrieve d at a given c, the higher is $r(d)$. The o_q coefficient represents the importance of the query. If we have a real user log, then this coefficient can be the likelihood of using the query; this relates to the number of times the query was issued by users. In our analysis, we consider $o_q = 1$ for all queries as the queries were simulated from the collection, not issued by real users.

In order to quantify the global retrievability bias across all documents in the collection, we follow [7] in using the Lorenz Curve [26] and the Gini Coefficient (G) which was proposed to summarize the bias in the Lorenz Curve [26]. If a system imposes no bias on the collection and all documents are equally retrievable, then $G = 0$. On the other extreme, if $G = 1$, then the same document is always retrieved for every $q \in Q$ and the remaining documents in the collection are never retrieved. The Lorenz Curve curve visually shows the retrieval bias variation between the retrieval models. The more the curve of a retrieval model deviates from the linear line of equality, the greater the bias imposed by that retrieval model.

4.2 Known-item search setup based on retrievability scores

In our known-item search experiment, a query formulated from a document (target document) is used to find that document, and the Mean Reciprocal Rank (MRR) is computed based on the position of the target document. In order to validate the relation between a document's retrievability score and the difficulty level of finding that document, we split documents into bins after sorting them based on their retrievability scores. We perform a known-item search experiment on the results of *BM25*, based on the two query sets, Q_a and Q_s. We select a high c, as more documents were retrieved ($r(d) > 0$); precisely we select $c = 100$, and $c = 1000$. Based on the Q_s query set, *BM25* retrieved 50.2% of the documents in collection at $c = 100$, and 71.8% at $c = 1000$. Based on the Q_a query set, 34.7% was retrieved at $c = 100$, and 64.9% was retrieved at $c = 1000$ by *BM25*. We perform the experiment based on the following steps:

1. Based on the documents' retrievability scores, we divide the collection into 4 bins. In addition to the retrieved documents, we include the non-retrieved ($r(d) = 0$) as they are the most difficult to retrieve.

   1a Azzopardi et al. sort the documents in ascending order based on their retrievability scores and divide them into 4 bins [7].

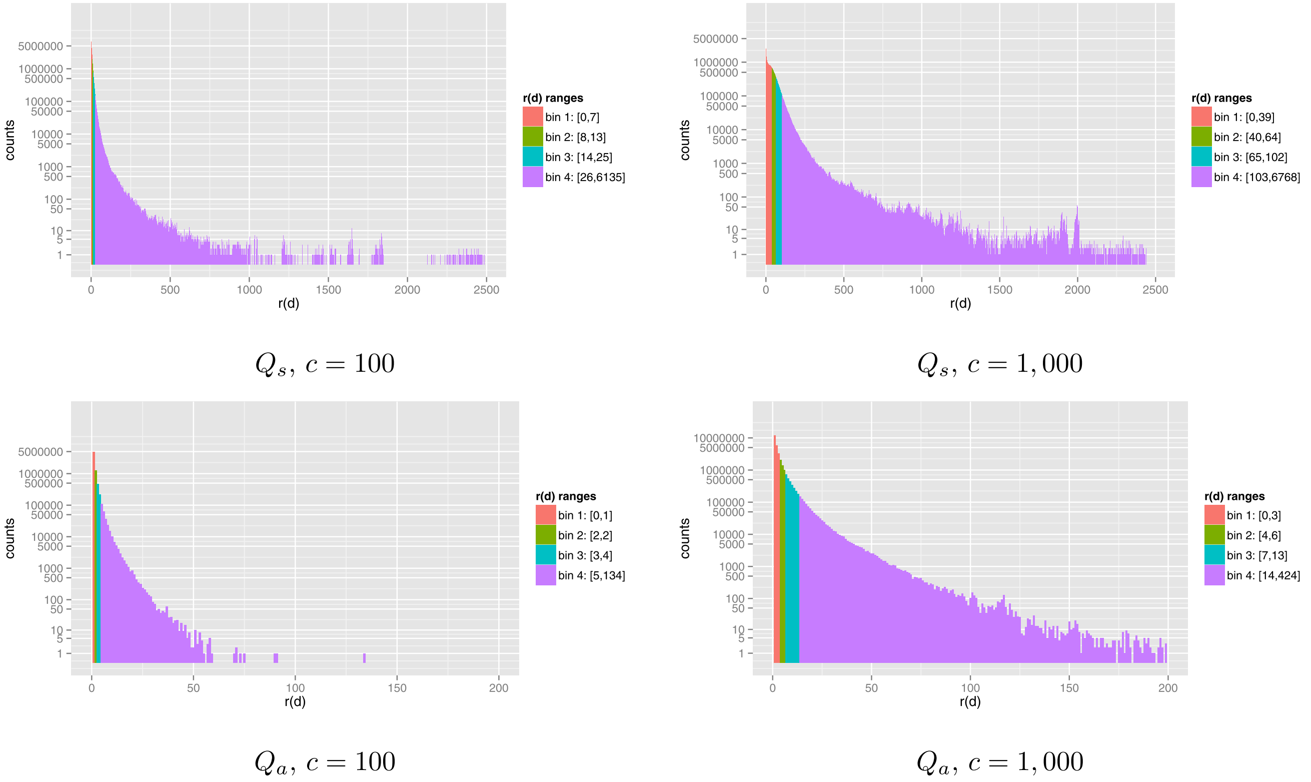
   1b In our setup this way of binning would mean that the non-retrieved documents dominate the first bins. The fraction of non-retrieved documents at $c = 100$ is 49.8% for the Q_s query set, which would mean that two bins would contain only those. The percentage of ($r(d) = 0$) based on the Q_a is higher, 35.1% at ($c = 1000$) and 65.3% at ($c = 100$). Instead, we chose to partition the documents based on the *wealth* distribution. The wealth is computed by multiplying each retrievability score by the number of documents having that retrievability score. We accumulate the wealth until 25% of the total wealth is reached, and we assign the corresponding documents to the bin. Figure 2 shows the values of documents' retrievability scores that contribute to the wealth of each bin, e. g., the first bin based on the Q_s query set at $c = 100$ contains all documents whose retrievability score is between 0 and 7.

2. From each bin, we randomly pick 1000 documents. Then, we formulate a query from each document, with a randomly chosen length between 3 and 7 terms. Then, terms that formulate the query were picked from the most frequent terms in the document until we get the required length. Stopwords, terms with less than 3 characters or a document frequency less than 2, and terms that occur in more than 25% of the documents in the collection are excluded. Finally, we issue these queries against the index of the whole collection using *BM25*.

5 Retrievability bias

First, we examine whether the search results obtained using three retrieval models on a Web archive collection are biased (*RQ1*) and investigate the extent of this bias. For this analysis, we assumed that a user is looking for an exact version of a document $d_v^{t_i}$. Every document's version was considered as a separate document at indexing time, and thus the relevance granularity was computed at the document's version granularity.

To compare the bias within the different result sets, we computed the *Gini Coefficients* for each query set, the three models and at different cutoff values (see Table 5). At $c = 10$, the Gini Coefficients are very high. For example, $G = 0.96$, $G = 0.95$, and $G = 0.96$ for *TF*IDF*, *BM25* and *LM1000*,



Q_s, $c = 100$

Q_s, $c = 1,000$

Q_a, $c = 100$

Q_a, $c = 1,000$

Fig. 2 Distribution of retrievability scores $r(d)$ for *BM25* based on all documents in the collection

Table 5 *Gini Coefficients* for all retrieval models with different values of c; all documents in the collection are used for computing the Gini Coefficient

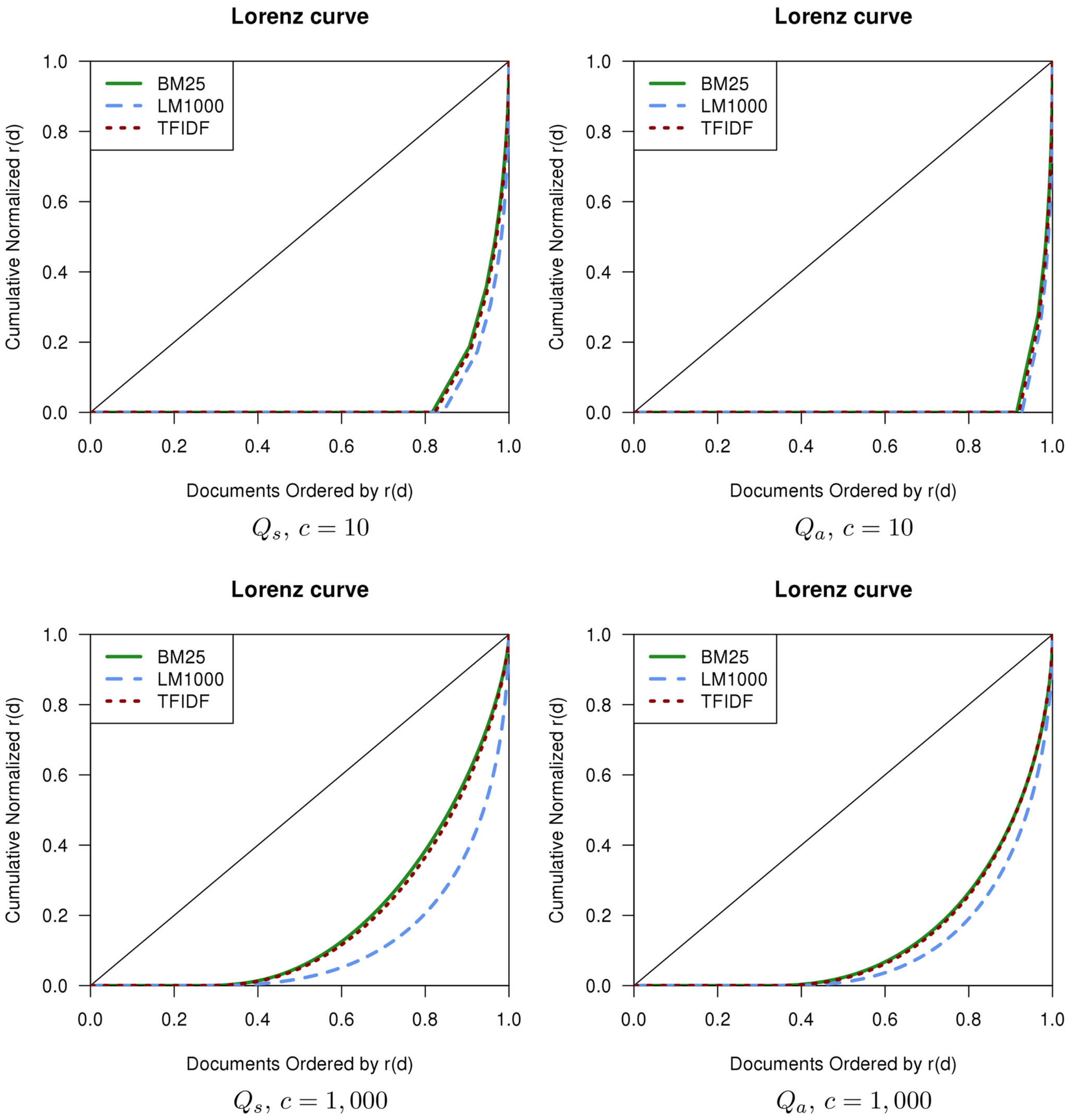
Query set	Ret. model	c		
		10	100	1000
Q_a	TFIDF	0.96	0.86	0.73
	BM25	0.95	0.85	0.73
	LM1000	0.96	0.88	0.79
Q_s	TFIDF	0.91	0.78	0.65
	BM25	0.90	0.76	0.63
	LM1000	0.93	0.84	0.77

The retrievability score was computed based on the document version granularity

respectively, based on the Q_a query set. These values are close to total inequality ($G = 1$). For higher values of c, the Gini Coefficients decrease. This trend is the same for the three models using the two query sets. However, even for $c = 1000$, the Gini Coefficients are still high. The least bias is found in the combination of *BM25* and the Q_s query set at $c = 1000$ ($G = 0.63$). The largest bias is induced by *LM1000* using the Q_a query set at $c = 10$ ($G = 0.96$). The differences concerning the extent of retrieval bias between the retrieval models, and between different values of c, are visualized in Fig. 3. *BM25* induces the smallest inequality for both query sets and can therefore be considered to be the fairest model. This is in line with the findings of [7,43].

For each setup a number of documents in the collection are never retrieved by any retrieval model ($r(d) = 0$). For the Q_a query set at $c = 10$, only 8% of the documents in the collection were retrieved by *TF*IDF*, 7.3% by *LM1000*, and 8.5% by *BM25*. The large fraction of documents that were *not retrieved* has a strong influence on the high values of the Gini Coefficients. This effect can be seen in the flat line of the Lorenz Curves for all c's. For example the Lorenz Curve of *BM25* at $c = 10$ deviates more from the equality line compared to the curve at $c = 1000$ and has a longer flat line.

Figure 4 shows the Lorenz Curve when the documents in the *3Models_union_c* were only considered for computing the bias. The deviation from the equality line across the models still has the same trend as in the case when all documents were considered. However, the deviation is smaller. Table 6 shows the Gini Coefficient for all models based on the documents in the *3Models_union_c* set. We cannot directly compare the Gini Coefficient values across the c's as they have been computed with different set sizes. However, we can still compare the models against each other at the same c, for example, we find that the *BM25* model induces the least inequality for both query sets at all values of c.



Q_s, $c = 10$

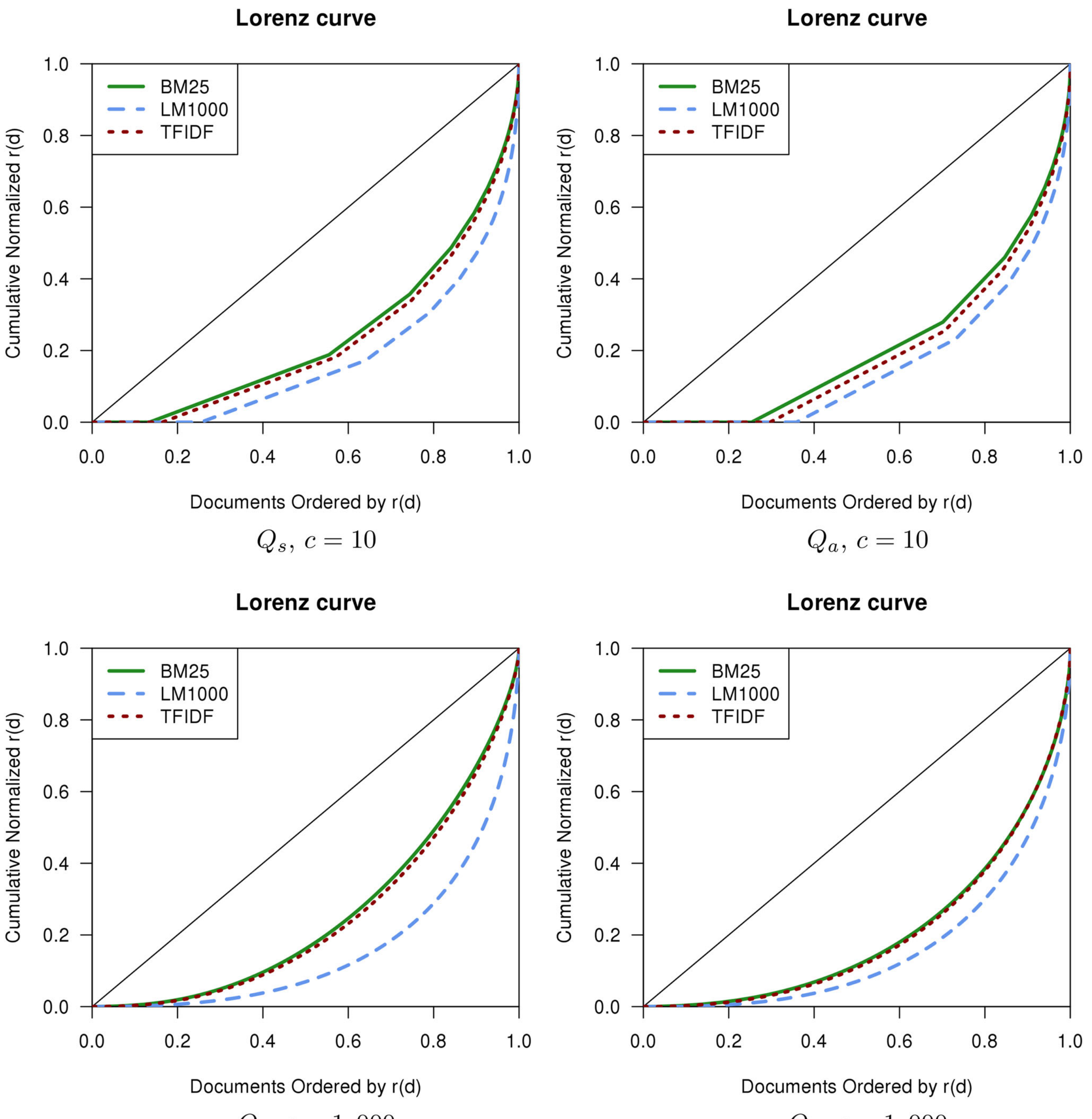
Q_a, $c = 10$

Q_s, $c = 1,000$

Q_a, $c = 1,000$

Fig. 3 *Retrievability scores inequality* among documents in the *entire collection* visualized with Lorenz Curve

We are interested in how many of the documents have a chance to be found using the models. The percentage of retrieved documents is the fraction of *unique* documents retrieved using *any* of the three models at a given c to the total number of documents in the collection. As c increases, more documents are retrieved (see Table 6). For example, based on the Q_a query set results, approximately 11% of the documents were retrieved using at least one model at $c = 10$; the remaining (89%) were *not retrieved* at all. The documents retrieved with *BM25* show the highest overlap with the *3Models_union_c* at different c's. For example, considering the *3Models_union_c* set created at $c = 10$, the percentage of overlap between the set of retrieved documents using the *BM25* model and the *3Models_union_c* set equals 75% (for query set Q_a) and 87% (for Q_s). On the other hand, for *LM1000* these percentages equal 64% and 75%, respectively.



$Q_s,\ c = 10$

$Q_a,\ c = 10$

$Q_s,\ c = 1,000$

$Q_a,\ c = 1,000$

Fig. 4 *Retrievability scores inequality* among documents in the *3Models_union_c* visualized with Lorenz Curve

5.1 Retrievability and findability

We explore the relation between the retrievability score and the findability of a document. We test the hypothesis in [7] which states that the lower the retrievability score of a document, the more difficult it should be to find it, even if the query is tailored to retrieve the target document. We use the known-item search setup as described in Sect. 4.2 to validate this hypothesis.

We computed the Mean Reciprocal Rank (MRR) to measure the effectiveness of the queries from each bin (see Table 7). We compare the MRR distributions of the first three bins with the fourth bin and test whether the differences between the bins are significant using the Kolmogorov-Smirnov test. We found that the bins with higher retrievability

Table 6 *Gini Coefficients* for all retrieval models with different values of c; document versions in the *3Models_union_c* at the corresponding c are considered for computing the Gini Coefficient

Query set	Ret. model	c						
		10	20	30	40	50	100	1000
Q_a	TFIDF	0.61	0.62	0.63	0.63	0.63	0.64	0.60
	BM25	0.57	0.59	0.60	0.60	0.60	0.61	0.59
	LM1000	0.67	0.69	0.69	0.70	0.70	0.71	0.68
	% retrieved union	11.4	17.9	22.7	26.4	29.4	39.2	66.3
Q_s	TFIDF	0.55	0.56	0.57	0.57	0.57	0.57	0.51
	BM25	0.53	0.54	0.54	0.55	0.55	0.55	0.49
	LM1000	0.65	0.67	0.68	0.68	0.69	0.70	0.69
	% retrieved union	21.1	30.8	36.6	40.6	43.6	52.0	72.4

The % retrieved is the fraction of retrieved document versions using the models from the whole collection at corresponding c

Table 7 Effectiveness of known-item queries measured by MRR

Query set	Rank cutoff c	Bins			
		1st	2nd	3rd	4th
Q_a	$c = 100$	0.12	0.35	0.37*	0.40
	$c = 1000$	0.12	0.25	0.25	0.31
Q_s	$c = 100$	0.09	0.30*	0.31*	0.30
	$c = 1000$	0.07	0.23*	0.25*	0.24

The first bin consists of the least retrievable documents, while the fourth bin contains the most retrievable documents. An $*$ indicates that the difference between the corresponding bin and the fourth bin is not significant using the Kolmogorov-Smirnov (p > 0.05)

scores also have a higher mean MRR score. The largest difference in the MRR distributions is between the first bin and the fourth bin for the two query sets and for both $c = 100$, and $c = 1000$. Using the Kolmogorov-Smirnov test, we can confirm that it is significantly easier to find documents from the fourth bin compared to documents from the first bin. This confirms our hypothesis and is in line with the findings presented in [4].

6 Impact of number of versions on the retrievability bias

In Sect. 5, we showed that all retrieval models impose a retrievability bias on the Web archive collection when we use the document's version as the basis. In this section, we explore the effect of varying numbers of versions of the same document on the retrievability bias (*RQ2*). First, we show how collapsing similar versions of the same document based on content similarity influences the retrieval bias (Sect. 6.1). Then, we use the number of versions per document to refine the search results after linearly combining a prior based on the number of versions with a score given using the retrieval model. In this approach, we collapse versions of the same documents based on their URLs (Sect. 6.2).

6.1 Collapsing similar versions

We first consider as a successful retrieval when the system returns any version of a specific document. In this scenario, the retrievability score of a document is computed by aggregating the retrievability scores of its versions. In a second scenario, we take the view that the content of the document's versions may have changed over time. Therefore, we cluster versions of the same document based on the similarity of their content, and we aggregate the retrievability scores at the cluster level. We believe that this experiment can be helpful when deciding which version(s) of a document to show to the user in the result lists as it allows other documents to appear in the top of the ranked results. We base the following experiments on the document's versions retrieved using the models and using the two query sets (discussed in Sect. 5).

Any version In this experiment, we consider finding any version of a document d at a given c a success. We compute the retrievability score $r(d)$ of a document d by accumulating the retrievability scores of its versions $r(d_v^{t_i})$. In the previous section, the retrievability scores were computed for document versions. In order to compute the retrievability score for documents, we map every document's version identifier to its URL. After that, we compute the Gini Coefficients for the three models with different c based on the documents in the union (see Table 8).

We found that the aggregation at document level increases the inequality bias for all retrieval models at all c's for the two query sets. We can derive that from the comparison of the Gini Coefficients in Table 8 with those in Table 6 (when the retrievability scores were computed at document version granularity). This can be explained by the varying number of versions per URL. On average every document is represented with 1.8 versions in the collection (see Table 1). Documents with a higher number of versions obtain higher retrievability scores as their versions are likely to appear multiple times in the ranked results at a given c. A similar trend exists for the

Table 8 *Gini Coefficients* for all retrieval models based on the two query sets, *Any version*

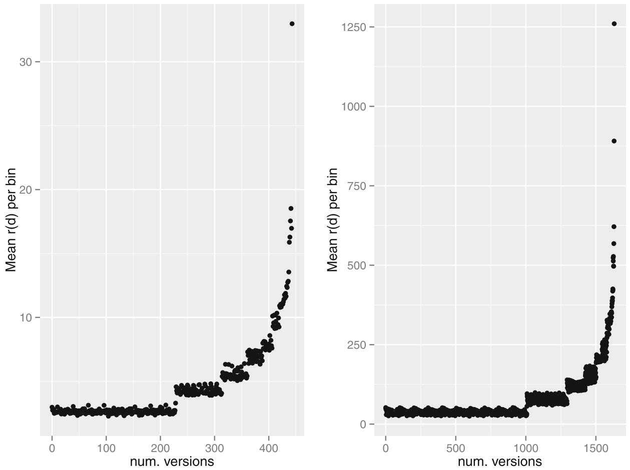
Query et	Ret. model	c						
		10	20	30	40	50	100	1000
Q_a	TFIDF	0.68	0.70	0.70	0.71	0.71	0.71	0.69
	BM25	0.66	0.67	0.68	0.68	0.69	0.69	0.69
	LM1000	0.74	0.75	0.75	0.76	0.76	0.76	0.75
	% retrieved union	11.7	17.3	21.4	24.7	27.3	36.0	62.5
Q_s	TFIDF	0.64	0.65	0.66	0.66	0.66	0.67	0.61
	BM25	0.62	0.63	0.64	0.65	0.65	0.65	0.60
	LM1000	0.71	0.73	0.74	0.74	0.75	0.75	0.73
	% retrieved union	21.2	29.3	34.2	37.7	40.3	48.0	68.9

other models, and also for the Q_s query set. In order to find out whether documents with a higher number of versions obtain higher retrievability scores, we plot the number of versions vs. retrievability scores. We did that by first sorting documents based on their number of versions and dividing them into bins. Each bin consists of 20,000 documents. For each bin, we calculated the mean retrievability score. We found that as the number of versions increases, the retrievability score increases as well (see Fig. 5).

Clustering versions (content-based similarity) In the previous experiment, we showed that the inequality increases when the $r(d)$ was computed at document (URL) granularity, by aggregating the retrievability score of all versions of the same document. As a next step, we explore the effect of grouping the most similar versions of the same document into two clusters. For every document in the Web archive collection, we first collect all versions of that document. We create a term frequency vector for each version and compute the cosine similarity between the versions. Finally, we split them into two clusters based on their similarity. We modify the retrieved results by the models by replacing the document's version identifier with the corresponding cluster identifier. Based on the mapping between document's version and cluster IDs, we compute the retrievability scores for every cluster.

Table 9 shows the Gini Coefficient for all retrieval models based on the Q_a and the Q_s query sets. Comparing the Gini Coefficients with those in Table 8 shows that the bias is smaller in the case of clustering compared to the *any version* case. Also the percentage of retrieved cluster IDs in the union of all models at given c is higher than the percentage of retrieved versions in the union at the corresponding c.

The Lorenz Curves show that the least bias is found when the retrievability score was computed at the document's version level (see Fig. 6). The bias increases when the retrievability score was computed at the document's level considering the two scenarios; the red and the blue curves are more deviated from the equality line. The bias is less based on the clustering of similar versions (red curve) compared to any match (blue curve); the difference is bigger at higher c.



Q_s, $c = 10$ Q_s, $c = 1000$

Fig. 5 Number of versions vs. retrievability score, for the *BM25* model

6.2 Collapsing versions (URL-based)

We showed that multiple versions of the same document impact the retrievability bias. This bias was the highest when the retrieval granularity was the document's version. In this section, we investigate the change in the retrieval bias when all versions of the same document are merged into one entry in the search result list based on their URLs. However, we take the number of versions into account for ranking documents, by embedding a prior based on the number of versions, with the retrieval models.

When a query q is issued, the retrieval model is used in computing a score (IR_{score}) for each document d in the collection based on how relevant its content is to the query q. Then the documents are ranked based on their relevance scores.

Including the temporal aspect of Web archives into retrieval models was discussed in [21]. In their model, they linearly combined a prior which favors documents with more

Table 9 *Gini Coefficients* for all retrieval models based on the two query sets, *Cluster version*

Query set	Ret. model	c						
		10	20	30	40	50	100	1000
Q_a	TFIDF	0.69	0.70	0.70	0.70	0.70	0.70	0.65
	BM25	0.67	0.67	0.68	0.68	0.68	0.68	0.65
	LM1000	0.75	0.76	0.76	0.76	0.76	0.76	0.72
	% retrieved union	18.6	27.5	33.7	38.5	42.3	54.0	81.5
Q_s	TFIDF	0.64	0.64	0.64	0.64	0.64	0.64	0.58
	BM25	0.62	0.62	0.62	0.62	0.62	0.62	0.57
	LM1000	0.72	0.73	0.74	0.74	0.74	0.74	0.72
	% retrieved union	34.2	45.8	52.4	56.7	59.9	68.5	87.3

versions or longer existence (time span between first version and last version) with known IR models. They showed that this approach achieved significant improvement over the baseline IR model.

We copy their approach and linearly combine the relevance score given to a document using a retrieval model (IR_{score}) with a score based on the number of versions for that document using the following formula:

$$IR_{score}^{versions} = \lambda * IR_{score} + (1 - \lambda) * prior_{versions} \qquad (2)$$

where IR_{score} is the relevance score as computed using the retrieval model for a document d and a given query q. The $prior_{versions}$ is a prior based on the number of versions; this prior is independent from the retrieval model. The value of this prior increases with the number of versions and is computed as follows:

$$prior_{versions} = \frac{\log_{10}(\#Versions)}{\log_{10}(max.\#Versions)} \qquad (3)$$

The number of versions per document is divided by the maximum $\log(max.\#Versions)$ in order to normalize the values to range from 0 to 1. We also normalize the values of IR_{score} given using the models to the same range. The retrieved documents are ranked from 1 to 1000 using the retrieval model for each query, and every document is assigned a score (IR_{score}). If the same document appears multiple times, then we take the maximum score. We adjusted the search results for each query by computing and sorting documents based on the new scores using Eq. 2. Finally, we computed the retrievability score using the documents in *3Models_union_c*.

We compared the Gini Coefficients of this experiment (Table 10) with results obtained by accumulating the retrievability scores of versions of the same document (Sect. 6; Table 8). We found that the inequality decreases for all models at all c's. This means that collapsing the versions of the same document reduces the retrievability bias induced by all models. However, the bias is still high, with the Gini Coefficient in the range between 0.51 and 0.75.

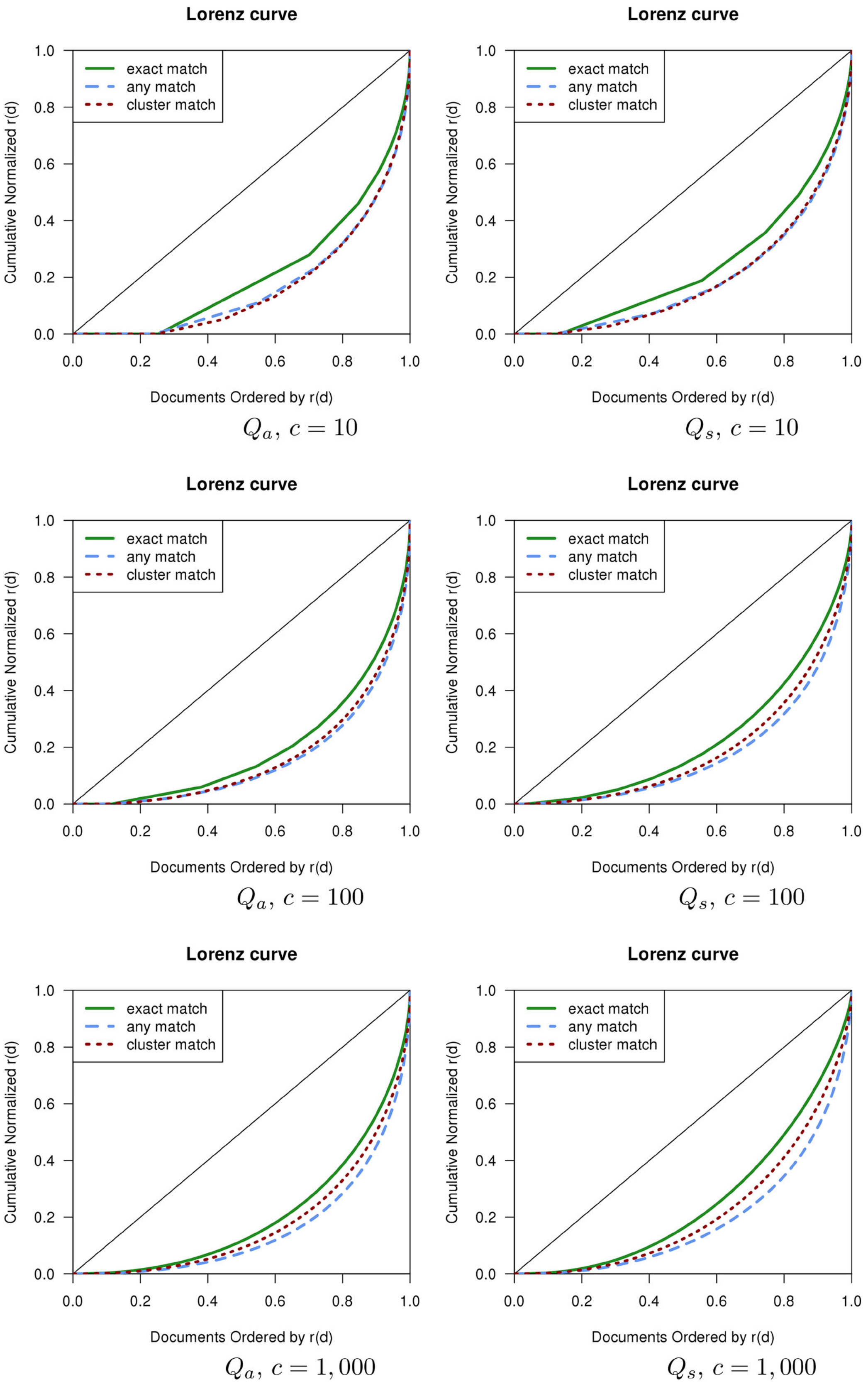
The percentage of retrieved documents increases because the retrieved items in the search results are the documents instead of the document's versions. We see a similar pattern for all values of c until we reach 1000; the percentage decreases as it approaches the maximum number of documents retrieved for the query. The difference in percentage retrieved in this experiment and the *any match* case increases as c increases.

7 Quantification of retrieval bias over the years

We investigated how the bias imposed by the retrieval system correlates with the number of documents aggregated over the years (*RQ3*). The Web archive collection consists of several crawls accumulated over time, and the number of websites included in the crawling process increased over the years. Therefore, the number of crawled documents varies. We explore whether the number of documents crawled in 1 year has an impact on the number of documents retrieved. For this experiment, we focused on *BM25* as it induced the smallest bias (see Sect. 5).

As mentioned in Sect. 4.1.1, every document's version in the Web archive collection has an associated crawling timestamp. We used this timestamp to divide the retrieved documents according to the year in which they were archived. This led to four subsets, 2009, 2010, 2011, and 2012. We apply the time-based splitting using the retrievability scores of documents computed for *BM25*, using the two query sets at different values of c: c = 10, 100, and 1000.

For every subset, we computed the mean retrievability score. We did not find a relation between mean retrievability score and subset size (see Table 11). The result is in line with [7]; for subsetting based on website domains, they found that there is no relation between subset size and the mean retrievability score computed per domain subset. As



$Q_a,\ c = 10$

$Q_s,\ c = 10$

$Q_a,\ c = 100$

$Q_s,\ c = 100$

$Q_a,\ c = 1,000$

$Q_s,\ c = 1,000$

Fig. 6 Lorenz curves visualizing the inequality of retrievability scores induced by *BM25* for three scenarios; exact match, any match, and cluster match using the anchor text (Q_a) and simulated (Q_s) query sets

Table 10 *Gini Coefficients* for the three retrieval models based on the two query sets, after embedding the prior based on number of versions with content similarity weight

Query set	Ret. model	c						
		10	20	30	40	50	100	1000
Q_a	TFIDF	0.64	0.65	0.66	0.66	0.67	0.67	0.62
	BM25	0.62	0.63	0.64	0.65	0.65	0.66	0.61
	LM1000	0.73	0.74	0.75	0.75	0.75	0.75	0.70
	% retrieved union	13.8	20.4	24.8	28.2	30.9	39.7	60.9
Q_s	TFIDF	0.60	0.61	0.62	0.62	0.62	0.62	0.53
	BM25	0.58	0.60	0.60	0.61	0.61	0.61	0.51
	LM1000	0.70	0.72	0.73	0.73	0.74	0.74	0.70
	% retrieved union	24.5	33.0	38.1	41.6	44.2	51.8	66.6

Table 11 *Retrievability subset analysis* using *BM25* results

	Q: Q_a									Q: Q_s								
Subset (size)	$c = 10$			$c = 100$			$c = 1000$			$c = 10$			$c = 100$			$c = 1000$		
2009 (12,232,831)	5.5	2.2	9.4	23.6	4.8	9.8	47.3	23.4	10.5	12.5	2.7	9.9	34.9	7.9	10.0	53.8	39.4	10.8
2010 (22,596,291)	7.9	2.1	24.7	33.0	5.0	25.4	63.0	25.2	25.9	17.3	2.6	25.2	48.6	7.8	25.8	71.4	39.5	26.5
2011 (30,275,150)	8.9	2.1	37.3	37.0	5.0	38.1	68.4	25.9	37.7	19.3	2.5	37.7	53.6	7.9	38.2	75.1	41.1	37.4
2012 (19,464,431)	10.6	2.2	28.6	40.4	5.4	26.7	73.1	28.1	25.9	21.7	2.6	27.3	56.9	8.3	26.0	79.0	43.8	25.3

For every subset, query set, and c: We present the fraction of retrieved documents from the subset in percentage (num. retrieved/subset size) (*first column*). The mean retrievability score of retrieved documents (*second column*). The fraction of retrieved documents from the corresponding subset to the number of all retrieved (num. retrieved per subset/all retrieved (all subsets)) in percentages (*third column*); sum of the percentage in this column is equal to 100%

expected, we found a relation between subset size and the percentage of retrieved documents. The larger a subset is, the higher the percentage of retrieved documents. For every subset, we computed the fraction of retrieved documents at given c, where the subset size is the same for all c's. The percentage of retrieved documents increases over the years until 2011, then drops for 2012 (see Table 11). We can explain this behavior by the number of documents that were crawled in each year. For example, the largest number of documents was crawled in 2011, and the highest percentage retrieved using *BM25* at all c's is from that same year.

Table 12 Summary of query subsets of Q_a query set

Q_a subsets	# of queries	Mean (query length)	# of terms
Q_a *_2009*	358,745 (N/A)	2.3	201,198
Q_a *_2010*	664,678 (69.0%)	2.4	326,725
Q_a *_2011*	998,350 (59.1%)	2.4	475,590
Q_a *_2012*	848,999 (41.9%)	2.4	411,263

For each subset, we show the number of queries. In parentheses is the number of unique queries in the corresponding subset (year) compared to previous years. For example, the Q_a *_2012* is compared against 2009, 2010, and 2011. For the 2009 subset the percentage of unique anchor texts is *N/A* as it is the first, and the percentage decreases across the years

7.1 Time-based subsets based on time-based queries

By binning the retrieved documents by year, we showed that the percentage of retrieved documents from a particular subset correlates with the number of documents in the bin. This analysis was based on simulated queries. Therefore, the number of queries we extracted from 1 year is directly linked to the number of documents that were crawled in that same year.

We further explore the relation between the queries' timestamps and the documents' timestamps. We focused our analysis on Q_a because the anchor texts are known to be a good substitute for both documents' titles and real queries. Recall that in Sect. 4.1.3, we generated the Q_a query set with a timestamp for each query which represents the crawling date. We divided the queries into 4 subsets, one for each year. We refer to these query sets as Q_a *_YYYY*, e.g., Q_a *_2009* represents anchor text extracted from links that were extracted from pages crawled in 2009. The number of anchor texts increases over years (see Table 12), but then drops for 2009. Because some documents exist in multiple versions, we expected to have overlapping anchor texts across the subsets. Therefore, along with the number of queries, we also show the number of unique queries per year compared to all previous years (see Table 12). For example, 41.9% of anchor texts from 2012 are new; they did not exist in any year before. The average query length is almost the same for all subsets (see Table 12). The distribution of query lengths is the same over the years (see Table 13).

Table 13 Query length distribution in the Q_a query set per year

Query length	2009	2010	2011	2012
1	26.2	23.2	23.7	23.9
2	33.5	34.9	34.1	34.4
3	24.2	24.9	24.9	23.5
4	11.8	12.7	12.9	13.0
5	3.8	3.8	3.9	4.7
6	0.5	0.5	0.5	0.5

Q_a *_2011* has the highest vocabulary size (see Table 12). The number of queries in 2012 is less than the number of queries in 2011 because fewer documents were crawled in 2012 compared to 2011 (see Table 1). In total, there are 100,908 terms shared across the vocabulary of the four query subsets.

We repeated the retrievability assessment as discussed in Sect. 4.1.4, with the four query subsets. We issued every $q \in Q_a$*_YYYY* for all subsets against the index of the entire collection. The query subsets are generated from the Q_a query set. Therefore, in order to explore the influence of these query subsets on the bias, we used the documents in the *3Models_union_c* set and the Q_a query set. When we compare the Gini Coefficients for the three retrieval models, we see that *BM25* leads to the smallest bias for the four query subsets at all of the studied values of c (see Table 14). The percentage of retrieved documents has an effect on the extent of the retrieval bias for all retrieval models. For example, the Q_a *_2009* query set shows the highest inequality for all retrieval systems because it has the smallest percentage of retrieved documents, whereas the Q_a *_2011* shows the smallest bias and has the highest percentage of retrieved documents. The result of this experiment confirms a relation between retrieval bias and number of documents crawled per year. We further investigate the relation between the timestamps of the queries and the timestamps of retrieved documents.

We performed a subset analysis based on the documents retrieved with *BM25* using the four query subsets, to measure differences in the retrieval bias over the years. For example, using the timestamps of documents retrieved using the *BM25* model using the Q_a *_2009* query subset, we partitioned the documents into 4 subsets, at different c's. For each subset and c we computed the mean retrievability score and the percentage of documents in that subset relative to the total, as we did in the subset analysis based on the Q_a. In addition to that, we computed the relative increase in the fraction of retrieved documents compared to running the Q_a query set (Table 11). This gives us an indication of how many documents we can retrieve from 2009 by running 2009 queries (Q_a *_2009*) compared to those we get by running queries from all years (Q_a). Running queries from a particular year causes the highest increase in the fraction of retrieved documents from that year (see Table 15). There is a relation between the timestamp of the queries and the timestamps of the documents. For example, using Q_a *_2009* at $c = 10$, 14.2% of retrieved documents using *BM25* originated from 2009, while by using entire anchor texts from all years (Q_a) at the same c, 9.4% of retrieved documents were from 2009. Running 2009 queries therefore results in a +4.8% increase

Table 14 *Gini Coefficients* for the three models at different c's using different query subsets, using documents in the *3Models_union_c* generated based on running the Q_a query set

Query Set	Ret. model	c						
		10	20	30	40	50	100	1000
Q_a *_2009*	TFIDF	0.85	0.84	0.82	0.81	0.81	0.77	0.64
	BM25	0.84	0.83	0.81	0.80	0.79	0.76	0.64
	LM1000	0.88	0.87	0.86	0.85	0.84	0.82	0.72
	% retrieved union	3.7	6.6	9.0	11.2	13.2	20.8	56.4
Q_a *_2010*	TFIDF	0.76	0.75	0.74	0.73	0.72	0.70	0.60
	BM25	0.74	0.73	0.72	0.71	0.70	0.68	0.60
	LM1000	0.80	0.79	0.78	0.78	0.77	0.76	0.68
	% retrieved union	6.2	10.5	14.0	17.0	19.5	28.8	62.0
Q_a *_2011*	TFIDF	0.70	0.69	0.69	0.68	0.68	0.67	0.60
	BM25	0.67	0.67	0.66	0.66	0.66	0.65	0.59
	LM1000	0.74	0.74	0.74	0.74	0.74	0.73	0.68
	% retrieved union	8.1	13.4	17.5	20.8	23.6	33.3	64.0
Q_a *_2012*	TFIDF	0.72	0.71	0.70	0.69	0.69	0.67	0.59
	BM25	0.69	0.68	0.68	0.67	0.67	0.65	0.59
	LM1000	0.76	0.76	0.75	0.75	0.75	0.74	0.68
	% retrieved union	7.4	12.4	16.2	19.5	22.2	31.8	63.4

Table 15 *Retrievability subset analysis based on time-aware queries* using *BM25* results

	Q: Q_a _2009		Q: Q_a _2010		Q: Q_a _2011		Q: Q_a _2012	
	Mean $r(d)$	%retrieved (%gain)	Mean $r(d)$	%retrieved (%gain)	Mean $r(d)$	%retrieved (%gain)	Mean $r(d)$	%retrieved (%gain)
$c = 10$								
2009	1.8	14.2 (**+4.8**)	1.7	9.9 (+0.5)	1.7	8.9 (−0.5)	1.6	8.7 (−0.7)
2010	1.5	26.0 (+1.3)	1.8	28.3 (**+3.5**)	1.7	23.9 (−0.8)	1.6	22.7 (−2.0)
2011	1.4	34.3 (−3.0)	1.6	35.8 (−1.5)	1.9	39.6 (**+2.3**)	1.7	36.4 (−0.9)
2012	1.4	25.5 (−3.0)	1.6	26.1 (−2.5)	1.8	27.6 (−1.0)	1.9	32.1 (**+3.5**)
$c = 100$								
2009	2.7	11.4 (**+1.6**)	2.9	9.8 (0.0)	3.3	9.5 (−0.3)	2.9	9.4 (−0.4)
2010	2.3	25.9 (+0.5)	3.1	26.4 (**+1.0**)	3.4	25.1 (−0.3)	3	24.7 (−0.7)
2011	2.1	36.8 (−1.3)	2.7	37.5 (−0.5)	3.7	38.6 (**+0.6**)	3.1	37.9 (−0.2)
2012	2.2	26.0 (−0.8)	2.9	26.2 (−0.5)	3.6	26.7 (0.0)	3.6	27.9 (**+1.2**)
$c = 1000$								
2009	7.4	10.8 (**+0.3**)	10.5	10.5 (0.0)	13.6	10.5 (−0.1)	11.7	10.4 (−0.1)
2010	6.7	25.7 (−0.2)	11.5	25.9 (0.0)	14.8	25.8 (−0.1)	12.6	25.7 (−0.2)
2011	6.4	37.4 (−0.3)	10.7	37.6 (−0.1)	16.1	37.8 (**+0.1**)	13.3	37.7 (0.0)
2012	6.7	26.1 (+0.2)	11.3	26.0 (**+0.1**)	16.2	26.0 (**+0.1**)	15.5	26.2 (**+0.3**)

Bold values represents running queries from a particular year causes the highest increase in the fraction of retrieved documents from that year

The fraction of retrieved documents per year to the total documents retrieved using *BM25* (*%retrieved*). The *%gain* represents the relative percentage of documents that we get per year using the corresponding query set to the % retrieved of the same year using the entire Q_a query set (Table 11)

of documents retrieved from that year. However, this effect decreases for higher c's.

8 Discussion and conclusions

In Web archives, the main focus has been on preserving the content from the Web before it is lost. Recently, Web archive initiatives started to make their Web archive collections available for search through full-text search systems so, as of yet, there are not many studies into the evaluation of Web archive search systems. The lack of queries with judged relevant documents for Web archives complicates such research. Retrievability has been proposed as an alternative that does not require relevance assessment, a measure that allows the quantification of accessibility bias. Retrievability has been applied in various studies on community-collected test collections such as the TREC collections. The documents in Web archives differ from these previously studied collections, however, because they are typically available in multiple versions which can be an implicit source of bias. We used the retrievability score per document and the overall bias measured by the Gini Coefficient and the Lorenz Curve of the retrievability scores of all documents to quantify the overall bias imposed by the retrieval model on the collection. We measured the retrievability and the overall bias in different scenarios in order to evaluate how the retrievability measure behaves under different retrieval models and different search scenarios. We also investigated whether search results in Web archives are influenced by varying numbers of versions, and how retrieval systems that are adapted to deal with them can be evaluated using retrievability.

We assessed the retrievability bias induced by three retrieval systems using retrievability scores, which we computed for each document's version in the collection. Our results show that the three systems induce bias at a document's version level, and there is a relation between the retrievability score of a document and the difficulty level of finding that document. Documents with higher retrievability scores are significantly easier to find, thus confirming that the retrievability score is a useful metric.

Then, we studied the change in bias when the system is adapted to deal with multiple versions of a document. We explored this using two approaches to collapse versions of the same document. First, we collapse document's versions based on their content similarity (*clustering-based*). Here, the cluster with more versions will get a higher retrievability score. Second, we collapse the versions based on their URL. Here, we embed a prior (based on the number of versions) with the scores given by retrieval systems; this means a document with more versions gets a higher score. The *clustering-based* approach takes into account that the content of document's versions may change over time, and thus collapse them into clusters. The *URL-based* approach considers them similar and collapses them into one URL. The bias was lower for the two collapsing approaches, as compared with the systems which do not consider the multiple versions of the document. The three retrieval systems impose lower bias in the URL approach, as compared to the clustering approach. We have shown that retrievability is suitable to assess Web archive retrieval systems, by showing its ability to capture the bias based on the approach followed to deal with multiple versions.

The evaluation of Web archives in terms of accessibility is important for both the institutions maintaining the archives and the users searching the archive. Knowing which documents are particularly hard to find allows the institutions to improve their retrieval systems and the users to adapt their search strategies and be aware of the retrieval bias and the source of that bias.

Acknowledgements This research was supported by the Netherlands Organization for Scientific Research (WebART project, NWO CATCH #640.005.001), and the Dutch COMMIT/program (SEALINCMedia project). We would like to thank the National Library of the Netherlands for their support. Part of the analysis work was carried out on the Dutch national e-infrastructure with the support of the SURF Foundation. We are grateful for the input from our colleagues Jiyin He and Desmond Elliott.

References

1. Alonso, O., Strötgen, J., Baeza-Yates, R.A., Gertz, M.: Temporal information retrieval: challenges and opportunities. TWAW **11**, 1–8 (2011)
2. Azzopardi, L., Bache, R.: On the relationship between effectiveness and accessibility. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 889–890. ACM (2010)
3. Azzopardi, L., de Rijke, M.: Automatic construction of known-item finding test beds. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06, pp. 603–604. ACM, New York (2006)
4. Azzopardi, L., de Rijke, M., Balog, K.: Building simulated queries for known-item topics: an analysis using six European languages. In: SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23–27, 2007, pp. 455–462 (2007)
5. Azzopardi, L., Vinay, V.: Accessibility in information retrieval. In: Proceedings of the Advances in Information Retrieval, 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30–April 3, 2008, pp. 482–489 (2008)
6. Azzopardi, L., Vinay, V.: Document accessibility: evaluating the access afforded to a document by the retrieval system. In: Workshop on Novel Methodologies for Evaluation in Information Retrieval, pp. 52–60. Citeseer (2008)
7. Azzopardi, L., Vinay, V.: Retrievability: an evaluation measure for higher order information access tasks. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, pp. 561–570. ACM, New York (2008)
8. Bache, R., Azzopardi, L.: Improving access to large patent corpora. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems II, pp. 103–121. Springer (2010)
9. Bashir, S., Rauber, A.: Analyzing document retrievability in patent retrieval settings. In: International Conference on Database and Expert Systems Applications, pp.753–760. Springer (2009)
10. Bashir, S., Rauber, A.: Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1863–1866. ACM (2009)
11. Bashir, S., Rauber, A.: Improving retrievability of patents in prior-art search. In: European Conference on Information Retrieval, pp. 457–470. Springer (2010)
12. Bashir, S., Rauber, A.: On the relationship between query characteristics and IR functions retrieval bias. J. Am. Soc. Inf. Sci. Technol. **62**(8), 1515–1532 (2011)
13. Ben-David, A., Huurdeman, H.: Web archive search as research: methodological and theoretical implications. Alexandria **25**(1–2), 93–111 (2014)
14. Berberich, K., Bedathur, S., Alonso, O., Weikum, G.: A language modeling approach for temporal information needs. In: European Conference on Information Retrieval, pp. 13–25. Springer (2010)
15. Callan, J., Connell, M.: Query-based sampling of text databases. ACM Trans. Inf. Syst. **19**(2), 97–130 (2001)
16. Campos, R., Dias, G., Jorge, A.M., Jatowt, A.: Survey of temporal information retrieval and related applications. ACM Comput. Surv. **47**(2), 15 (2015)
17. Costa, M., Couto, F., Silva, M.: Learning temporal-dependent ranking models. In: Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 757–766. ACM (2014)
18. Costa, M., Gomes, D., Silva, M.J.: The evolution of web archiving. Int. J. Digit. Libr. **17**, 1–15 (2016)
19. Costa, M., Silva, M.J.: Understanding the information needs of web archive users. In: Proceedings of the 10th International Web Archiving Workshop, vol. 9, p. 6 (2010)
20. Costa, M., Silva, M.J.: Characterizing search behavior in web archives. In: WWW2011 Workshop on Linked Data on the Web, Hyderabad, India, March 29, 2011, pp. 33–40 (2011)
21. Costa, M., Silva, M.J.: Evaluating web archive search systems. In: Proceedings of the Web Information Systems Engineering—WISE 2012-13th International Conference, Paphos, Cyprus, November 28–30, 2012, pp. 440–454 (2012)
22. Craswell, N., Hawking, D., Robertson, S.: Effective site finding using link anchor information. In: SIGIR, pp. 250–257. ACM (2001)
23. Dou, Z., Song, R., Nie, J.-Y., Wen, J.R.: Using anchor texts with their hyperlink structure for web search. In: SIGIR, pp. 227–234 (2009)
24. Eiron, N., McCurley, K.S.: Analysis of anchor text for web search. In: SIGIR 2003: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 28–August 1, 2003, Toronto, Canada, pp. 459–460 (2003)
25. Fujii, A.: Modeling anchor text and classifying queries to enhance web document retrieval. In: WWW, pp. 337–346 (2008)
26. Gastwirth, J.L.: The estimation of the lorenz curve and gini index. Rev. Econ. Stat. **54**(3), 306–316 (1972)

27. Gomes, D., Miranda, J., Costa, M.: A survey on web archiving initiatives. In: TPDL, pp. 408–420 (2011)
28. Gomes, D., Nogueira, A., Miranda, J., Costa, M.:Introducing the Portuguese web archive initiative. In: 8th International Web Archiving Workshop. Springer (2009)
29. Huurdeman, H.C., Kamps, J., Samar, T., de Vries, A.P., Ben-David, A., Rogers, R.A.: Lost but not forgotten: finding pages on the unarchived web. Int. J. Digit. Libr. **16**(3), 247–265 (2015)
30. Jin, R., Hauptmann, A.G., Zhai, C.: Title language model for information retrieval. In: SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11–15, 2002, Tampere, Finland, pp. 42–48 (2002)
31. Kamps, J.: Web-centric language models. In: CIKM, pp. 307–308 (2005)
32. Kanhabua, N., Blanco, R., Nørvåg, K.: Temporal information retrieval. Found. Trends Inf. Retr. **9**(2), 91–208 (2015)
33. Klein, M., Nelson, M.L.: Moved but not gone: an evaluation of real-time methods for discovering replacement web pages. Int. J. Digit. Libr. **14**(1–2), 17–38 (2014)
34. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
35. Koolen, M., Kamps, J.: The importance of anchor text for ad hoc search revisited. In: SIGIR, pp. 122–129 (2010)
36. Kraft, R., Zien, J.: Mining anchor text for query refinement. In: Proceedings of the 13th International Conference on World Wide Web, WWW, pp. 666–674. ACM, New York (2004)
37. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
38. Metzler, D., Novak, J., Cui, H., Reddy, S.: Building enriched document representations using aggregated anchor text. In: SIGIR (2009)
39. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120
40. Ras, M.: Eerste fase webarchivering. Technical Report, Koninklijke Bibliotheek (2007)
41. Ras, M., van Bussel, S.: Web archiving user survey. Technical Report, National Library of the Netherlands (Koninklijke Bibliotheek). https://www.kb.nl/sites/default/files/KB_UserSurvey_Webarchive_EN.pdf (2007)
42. Rauber, A., Bruckner, R.M., Aschenbrenner, A., Witvoet, O., Kaiser, M.: Uncovering information hidden in web archives: a glimpse at web analysis building on data warehouses. D-Lib Mag. **8**(12). http://www.dlib.org/dlib/december02/rauber/12rauber.html (2002)
43. Traub, M.C., Samar, T., van Ossenbruggen, J., He, J., de Vries, A., Hardman, L.: Querylog-based assessment of retrievability bias in a large newspaper corpus. In: Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, pp. 7–16. ACM (2016)
44. Wilkie, C., Azzopardi, L.: Relating retrievability, performance and length. In: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 937–940. ACM (2013)