Understanding Content Reuse on the Web: Static and Dynamic Analyses

Ricardo Baeza-Yates[1], Álvaro Pereira[2⋆], and Nivio Ziviani[2]

[1] Yahoo! Research &
Barcelona Media Innovation Centre
Barcelona, Spain
rbaeza@acm.org
[2] Department of Computer Science
Federal University of Minas Gerais
Belo Horizonte, Brazil
{alvaro,nivio}@dcc.ufmg.br

Abstract. In this paper we present static and dynamic studies of duplicate and near-duplicate documents in the Web. The static and dynamic studies involve the analysis of similar content among pages within a given snapshot of the Web and how pages in an old snapshot are reused to compose new documents in a more recent snapshot. We ran a series of experiments using four snapshots of the Chilean Web. In the static study, we identify duplicates in both parts of the Web graph – reachable (connected by links) and unreachable components (unconnected) – aiming to identify where duplicates occur more frequently. We show that the number of duplicates in the Web seems to be much higher than previously reported (about 50% higher) and in our data the duplicated in the unreachable Web is 74,6% higher than the number of duplicates in the reachable component of the Web graph. In the dynamic study, we show that some of the old content is used to compose new pages. If a page in a newer snapshot has content of a page in an older snapshot, we say that the source is a parent of the new page. We state the hypothesis that people use search engines to find pages and republish their content as a new document. We present evidences that this happens for part of the pages that have parents. In this case, part of the Web content is biased by the ranking function of search engines.

1 Introduction

The Web grows at a fast rate and little is known about how new content is generated. At the same time, a large part of the Web is duplicated. Other pages are created by using older pages, such as by querying a search engine, selecting a few highly ranked pages and copying selected paragraphs from them. In this paper we present static and dynamic studies involving the analysis of similar

⋆ This work was partially done when at Yahoo! Research Barcelona as a Ph.D. intern.

content among pages within a given snapshot of the Web and how pages in an old snapshot are reused to compose new documents in a more recent snapshot.

For the static study, we present an algorithm to find duplicate and near-duplicate documents in a Web collection. Considering each collection independently (statically), we obtained the frequency of duplicates in a set of documents that can be reached by following links. Our aim is to identify where duplicates occur more frequently on the Web graph and what the impact on coverage is when only crawling documents from the reachable set of the Web graph. We show that the Web has many more duplicates than previously acknowledged in the literature, because we had access to most of the unconnected part of the Web graph. For instance, the work in [1] used collections crawled by following links on the Web. In this case the sample of the Web is biased because most of the documents are crawled from the reachable set of the Web graph.

Considering the collections in an evolutionary way (dynamically), we show how old content is used to create new content. At first, we looked for original sources, if any, of the content of a new page. We can say that each source is a *parent* of a new page and hence we can study how old content evolves in time, that is, which pages are really new and do not have parents and which ones have parents. Additionally, we state the hypothesis that when pages have parents, most of the time there was a query that related the parents and made possible for a person to create the new page. If this is the case, some Web content is biased by the ranking function of some search engine.

We ran a series of experiments using four snapshots of the Chilean Web Considering our data set: i) we analyze if duplicates occur more or less frequently according to reachable and unreachable sets of the Web graph; ii) we present a study about the influence of old content in new pages; iii) we present evidence that search engine ranking algorithms are biasing the content of the Web; and iv) we show that the number of copies from previously copied Web pages is indeed greater than the number of copies from other pages.

This paper is organized as follows. Section 2 presents definitions and the Web collections used in the experiments. Section 3 presents an algorithm to detect duplicates and a static study on Web content reuse. Sections 4 and 5 present a dynamic study of our Web collections. Section 4 presents a study of the relation of search engines with the Web content evolution. Section 5 shows how much old content is used to compose new documents. Section 6 presents related work. Finally, Section 7 presents the conclusions of our work.

2 Conceptual Framework

In this section we present some definitions and the Web collections used in the experiments.

2.1 Definitions

The definitions are the following:

Definition 1. Shingle Paragraph*: A shingle paragraph is a sequence of three sentences of the document, where a sentence is a sequence of words ended by a period.* It is a way of measuring the content similarity among documents, using the concept of shingles [2]. If a period is not found until the 150th character, then the sentence is finished at that point and a new sentence begins at the 151th character. This limitation is due to the fact that some documents have no period (for example, some program codes). In this work we used two types of shingle paragraphs: **with overlap** of sentences and **without overlap** of sentences. As an example, suppose we have a document containing six sentences s_1. s_2. s_3. s_4. s_5. s_6, where s_i, $1 \leq i \leq 6$, is a sentence of the text. The shingle paragraphs with overlap of sentences are: "s_1. s_2. s_3.", "s_2. s_3. s_4.", "s_3. s_4. s_5.", "s_4. s_5. s_6.". The shingle paragraphs without overlap of sentences are: "s_1. s_2. s_3.", "s_4. s_5. s_6.".

Definition 2. Cluster*: For a given collection, it is a set of documents with exactly the same shingle paragraphs, without overlap of sentences. Each document in a collection is either (i)* ***clustered****, if it belongs to a cluster, or (ii)* ***unique****, otherwise.*

Definition 3. Duplicate Document*: It is any clustered document with exception of the original document that initiated the cluster.* Finding the original document is not important, we only need to consider that it exists in order to calculate the number of duplicates in a given collection. We can say that two documents of the same cluster are duplicates one of the other. **Near-Duplicate***: It is a document with a given minimal percentage of identical shingle paragraphs of another document in the collection.* This percentage is related to the number of shingle paragraphs of both documents.

Definition 4. Reachable Component*: For a given collection, it is a set of documents that can be reached by following links, from a given initial document.* **Unreachable Component***: It is the set of documents that are not in the reachable component.* The initial document is randomly chosen. If less than 50% of the documents are reached by following links from a given initial document, another document must be used as the initial document. The objective is to simulate a Web crawler following links (navigational and external links) and composing a Web database. Considering the macro structure of the Web proposed by Broder [3], the reachable component comprises the "central core" (the strongly connected component [4]). If the initial document belongs to the "in" component, a part of this component will also be included in the reachable component just defined. As well, the last reachable documents found belong to the "out" component.

Definition 5. Minimal Number of Identical Paragraphs*: It is a minimal threshold of the number of identical paragraphs to consider a new document (in a more recent collection) as a partial copy of an old document (in an older collection).*

Definition 6. New Similar Document*: It is a new document composed by at least one paragraph existent in an old document.*

Definition 7. Answer Set*: For a given query, it is the document set returned by the query processor of a search engine.* **Total Answer Set***: for a given query log, it is the document set composed by the union of the answer sets of all queries.*

Definition 8. Equivalent Documents*: Two documents in two distinct Web collections are equivalent if their URLs are identical.* In this case, a document in an older collection remains existing in a more recent collection.

Definition 9. Document Relationship*: A new document has a* ***parent*** *if it shares a minimal number of identical shingle paragraphs with the parent document and they are not equivalent. An old document has a* ***child*** *on the basis that it shares a minimal number of identical paragraphs with the child document and they are not equivalent.* These definitions are recursive if more than two collections are considered. Thus, for three collections it is possible to identify grandparents and grandchildren, considering the newest and the oldest collections, respectively.

2.2 Web Collections

For the experiments we used four collections of pages of the Chilean Web that were crawled in four distinct periods of time. Table 1 presents the main characteristics of the four collections.

Table 1. Characteristic of the collections.

Collection	Crawling date	# of docs (millions)	Size (Gbytes)
2002	Jul 2002	1.04	2.3
2003	Aug 2003	3.11	9.4
2004	Jan 2004	3.13	11.8
2005	Feb 2005	3.14	11.3

In our experiments we considered only documents with more than 450 characters and at least three shingle paragraphs with overlap of sentences (see Definition 1). In order to consider two documents as similar, previous experiments show that it is necessary to have a minimal degree of similarity between them to avoid finding too many false matches. This is the case when only one or two popular shingle paragraphs are identical. Following these restrictions, the number of documents considered from the collections presented in Table 1 is reduced to approximately 75% in our experiments.

Every collection was crawled by the Web search engine TodoCL[3]. To crawl the collections, the complete list of the Chilean Web primary domains were used to start the crawling, guaranteeing that a set of pages under almost every Chilean domain (.cl) was crawled, once the crawls were pruned by depth. These characteristics are fundamental for studying the evolution of the Web content.

[3] www.todocl.cl or www.todocl.com

3 Duplicate Study

In this section we present the duplicate study. Section 3.1 presents the algorithm we used to find duplicate and near-duplicate documents (see Definition 3) and Section 3.2 presents the results about duplicates for our collections.

3.1 Algorithm for Duplicate Detection

The algorithm works by clustering duplicate documents [5]. Since our collections are not large (see Table 1), the algorithm uses the whole text of the documents for comparison, improving the precision of the results.

The comparison step of the algorithm uses shingle paragraphs without overlap of sentences (see Definition 1). Collection C (with n documents) is divided into m subcollections S_i, $0 \leq i < m$. The algorithm runs in m steps. For each subcollection S_i, $0 \leq i < m$, the shingles of the documents in S_i are first inserted into a hash table.

Next, the shingles of C are searched in the hash table. A duplicate is detected if all shingles of a document in C have a match in a document of S_i and both documents have the same number of shingles. At the end of each iteration i, the subcollection S_i is excluded from C ($C = C - S_i$).

For each new duplicate pair found, a new cluster (see Definition 2) is created and the duplicate pair is inserted into the new cluster. For that, a cluster identifier is associated with each document. If one of the documents of the pair was previously inserted into a given cluster, then the other document of the pair is inserted into this cluster. At the end, the algorithm returns a set of clusters, with each cluster containing a list of clustered documents.

Figure 1 illustrates the main steps of the algorithm using a sample test collection C containing $n = 20$ documents. In the example, collection C is divided into $m = 10$ subcollections, each one containing two documents. Sentences in each document are represented by letters, as shown in documents 1, 2, 19 and 20. Every document contains four shingle sentences (for instance, document 1 has the shingles "*a. a. a.*", "*b. b. b.*", "*c. c. c.*", "*d. d. d.*").

Following Figure 1, in the first iteration, the documents 1 and 2 (from subcollection S_0) are inserted into the hash table. Next, the shingles of the documents of C (documents 1 to 20) are searched in the hash table. Therefore, it is possible to see that document 19 is a duplicate of document 2. In the second iteration, documents 3 and 4 (from subcollection S_1) are inserted into the hash table and the shingles of the documents of collection C (documents 3 to 20) are searched in the hash table. Next iterations occur similarly.

When using this algorithm, false matches occur when two documents have the same number of identical shingle paragraphs, but with some repeated shingle. For example, suppose that the document 3 in Figure 1 has the following sentences: *e. e. e. d. d. d. e. e. e. d. d. d* (the shingles are "*e. e. e.*", "*d. d. d.*", "*e. e. e.*" and "*d. d. d.*"). Once every shingle of the document 3 is found in the hash table for the document 2 and both documents have four shingle paragraphs, then they

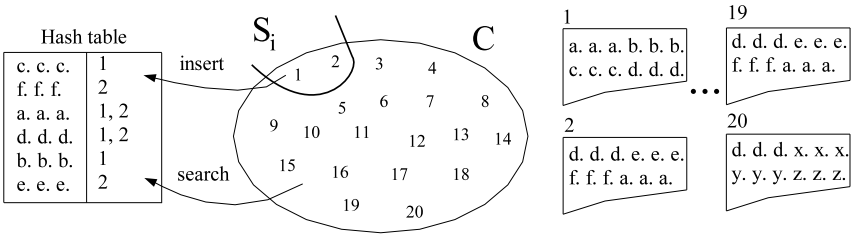


Fig. 1. Process for duplication analysis.

are considered duplicates. As this situation seems to occur with a very small probability, the percentage results are not biased by false matches.

3.2 Results about Duplicates

In this section we present the results about duplicates and near-duplicates. Table 2 presents statistical results for the collections used. According to the table, the number of clustered documents (that is, the number of clusters in addition to the number of duplicates) represents more than 50% of the documents for collections 2003 and 2004. For collection 2004, only 48.9% of the documents are unique (i.e., do not belong to a cluster).

Table 2. General statistics about duplicates.

Collection	# of docs	# of clusters	# of dup.	% of dup.
2002	614,000	76,000	196,000	31.9%
2003	2,067,000	252,000	804,000	38.9%
2004	2,033,000	266,000	876,000	43.1%
2005	2,175,000	256,000	778,000	35.8%

In turn, Table 3 shows the number of near-duplicates compared to the number of duplicates for each collection. We considered three values of minimal percentage of identical shingle paragraphs: 90%, 70% and 50% (see Definition 3).

Table 3. Data about near-duplicates.

Collection	% of dup.	90% near.	70% near.	50% near.
2002	31.9	33.4	35.7	39.9
2003	38.9	40.8	46.1	52.8
2004	43.1	44.5	47.9	53.4
2005	35.8	37.0	43.2	49.9

The percentage of near-duplicates for 90% of similarity is slightly greater than the percentage of duplicates. For instance, for collection 2003, only 1.9% of the documents share at least 90% of their shingles paragraphs and are not duplicates. On the other hand, for the same collection, 7.1% of the documents share at least 70% of their shingles paragraphs and are not duplicates, and

13.9% of the documents share at least 50% of their shingle paragraphs and are not duplicates.

Again, the analysis of Table 2 reveals that collection 2002 has the smallest percentage of duplicates (31.9%) whereas collection 2004 has the highest percentage (43.1%). These figures are higher than the figures found in the literature (Shivakumar and Garcia-Molina [6] reports 27% and Fetterly, Manasse and Najork [7] reports 22%).

Our hypothesis was that the difference occurs because our collections were crawled based on a list of primary domains, which includes URLs that cannot be reached following links obtained from other pages. Most of the Web crawlers work following links, considering only documents from the connected component of the Web graph. Duplicates do not have the same inlinks as the original document (the source of the duplicates).

To study this hypothesis we place the documents of each collection in either reachable or unreachable component, according to Definition 4. According to the number of documents, the reachable component represents 54.0%, 51.6%, 56.8% and 63.9% of the collections 2002, 2003, 2004 and 2005, respectively. Table 4 presents the number and percentage of duplicates for the complete collection, for the reachable component and for the unreachable component.

Table 4. Percentage of duplicates for the complete collection and, the reachable and unreachable components.

Collection	Complete		Reachable		Unreachable	
	number	perc. (%)	number	perc. (%)	number	perc. (%)
2002	196,000	31.9	62,000	18.9	108,000	38.2
2003	804,000	38.9	277,000	25.9	430,000	43.0
2004	876,000	43.1	339,000	29.3	443,000	50.4
2005	778,000	35.8	361,000	26.0	323,000	41.1

Observing Table 4 for the collection 2002 we see that the real number of duplicates for the complete collection is 68.8% higher than the number of duplicates found for the reachable component. On average this percentage is 50.9%, considering the four collections. The number of duplicates found for the unreachable component is on average 74.6% higher than the number of duplicates for the reachable component. These expressive percentages show that most duplicates occur in the unreachable component of the Web graph. We also verify that the absolute real number of duplicates is about two or three times the absolute number of duplicates in the reachable component. One reason for this results is that text spam might be more common in the unreachable part and one well used technique is to mix paragraphs from other pages.

To support our results we analyze the number of duplicates in a Brazilian Web collection [8] crawled in 1999 by the TodoBR search engine[4]. For this collection

[4] TodoBR is a trademark of Akwan Information Technologies, which was acquired by Google in July 2005.

a list of domains was **not** used to start the crawler. It means that new pages are normally found only when a link to these pages are added from an existing page in an older snapshot. The percentage of duplicates for this collection is very similar to what is acknowledged in the literature: 24.9% of the documents are duplicates. This supports our conclusion that including the disconnected component (such as for the Chilean collections), the Web has more duplicates than previously acknowledged in the literature.

Returning to the Chilean Web collection, now we study the relations between duplicate and cluster sizes. For collection 2004, nine clusters have more than 10,000 documents, in which two of them have more than 20,000 documents. The duplicates belonging to these clusters represent 7.1% of the documents of the collection. This explains the high number of duplicates for collection 2004 in relation to the other collections studied, as shown in Table 2.

For collection 2003, 95.7% of the clusters have ten or less documents. Documents in these small clusters represent 63.3% of clustered documents and only 40.5% of duplicate documents. The same figures were found for the other collections, which means that large clusters have more influence in the number of duplicates than small clusters.

Clusters with two documents (with only one duplicate) are very frequent. Collections 2002, 2003, 2004 and 2005 have 52,000, 158,000, 167,000, 160,000 clusters containing only two documents, respectively. In every case these values represent about 63% of the clusters, but only approximately 19% of the duplicates.

Our results on duplicates have an important impact for search engine Web crawlers. Once the Web grows at a very fast rate, is extremely dynamic and has many replicated content, search engines have to heuristically decide which pages to crawl. In this section we have shown that the reachable component of our Web graphs contain a representative portion of the Web, in terms of coverage. In order to design a Web crawler, many different aspects must be considered. Considering the coverage and elimination of duplicates, Web crawlers designers may choose to crawl only pages reached by links, instead of listing every found directory and crawling every document in a directory.

4 Log-Based Content Evolution Study

In this section we present an algorithm and experiments related to the log-based content evolution study. In this part of the work we mine data with the objective of supporting the hypothesis that people use search engines to find pages and republish their content as a new document. Section 4.1 presents a description of the algorithm. Section 4.2 presents the setup procedure to perform the experiments. Section 4.3 presents the experimental results for this study.

4.1 Algorithm Description

In this section we describe a log-based algorithm to study the evolution of the Web content. The algorithm is composed of two stages. The objective of the

first stage is to find new similar documents (see Definition 6), which are candidates of being copies. The objective of the second stage is to filter the new similar documents found in the first stage and find (with a high probability) new documents with content of old documents returned by queries. By finding those documents, we present evidence that states the initial hypothesis. The two stages are described in this section.

Finding New Similar Documents. We use Figure 2 as an example to explain the method to find new similar documents, with the purpose of finding candidates to be filtered in the second stage of the algorithm. For this, we consider pairs of old–new Web collections, referring to the older collection as *old* (O) and to the more recent collection as *new* (N). We explain the method dividing it into three main steps.

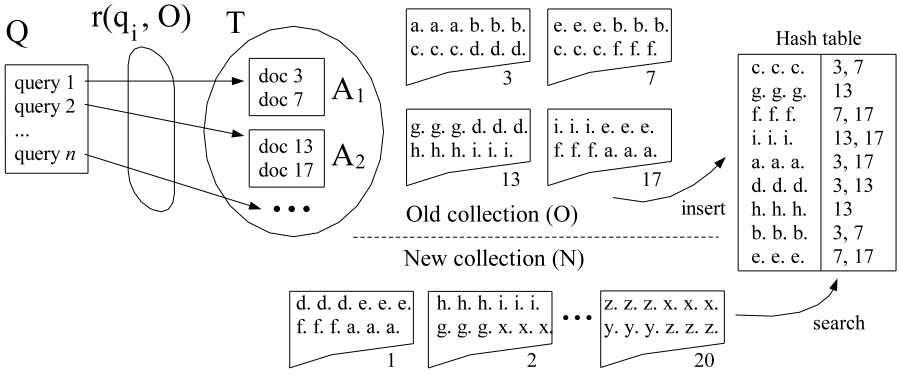


Fig. 2. Method to find new similar documents.

Firstly, a set Q of queries (a query log) is used to simulate a user performing a query on the search engine. The query processor of the search engine TodoCL is used as the ranking function and is applied to each query as well as to the old collection. An answer set A_i (see Definition 7) is returned for each query performed. In the example of Figure 2 the ranking function returns the documents 3 and 7 for the query 1 and the documents 13 and 17 for the query 2. The content of these documents are shown in the figure.

Secondly, each document from the total answer set T (see Definition 7) has its shingle paragraphs extracted and inserted into a hash table. We use shingle paragraphs with overlap of sentences (see Definition 1). With the purpose of comparison, shingles are normally used in samples, as a randomized technique that allows false positives. In this part of the work we consider **all** the shingle paragraphs of the documents, with the advantage of improving the precision.

Thirdly, each document from the new collection N has its shingle paragraphs searched in the hash table. A new similar document is detected when at least one shingle of the new document is found in the hash table. While new documents are being compared a table is constructed containing important data for the

next stage: the new similar document identifier, the old document identifier, and the query identifier.

In the example of Figure 2 the new collection has 20 documents (documents 1, 2 and 20 are shown). Document 1 is a new similar document, since one or more shingles of this document are found in the hash table. Document 1 has shingle paragraphs from documents 3, 7, 13 and 17. Document 2 is also a new similar document.

An important goal of this algorithm stage is the possibility of repeating the search engine operation in a given period of time. We are able to repeat what had been done in the past by users of the search engine TodoCL, recovering the same Web documents that were recovered on that period of time. This is possible because:

- We know the periods of time (with a good approximation) that every collection was indexed and used in the search engine (see Table 1).
- We used the same query processor used by the search engine in each period of time between every collection pair.
- We used the most frequent performed queries, aiming to increase the probability of finding a query used for copying by at least one of the users that performed that query in the past.

Filtering New Similar Documents. At this stage the new similar documents found in the first stage are filtered. Besides the data returned from the previous stage, the conditions to filter also use data about duplicates returned by the duplicate detection algorithm (see Section 3), and data with the URLs of the documents for every collection. The conditions to filter are the following:

1. Consider a minimal number of identical paragraphs (see Definition 5). We studied six minimal values: 5, 10, 15, 20, 25 and 30 identical shingle paragraphs. This condition is important to eliminate a false match, with only a few identical shingle paragraphs, that occurs because some documents have, for example, an identical prefix or suffix automatically generated by an html editor.
2. The new document must be composed by pieces of two old documents returned by the same query. It is intuitive that, in many cases, if the new document has some content of documents returned by the same query, a user might have performed that query before composing the new document. We think that in many cases a user performed a query and used only one query result to compose a new page. This situation cannot be captured by our algorithm. If we considered this situation, we could not infer that the user found that page because he/she previously performed the query in the search engine.
3. The new document must contain at least two **distinct** shingle paragraphs from each old document, in order to ensure that the old content used in the new document is not the same amongst both of the two old documents.

4. The new document URL cannot exist in the old collection. This condition guarantees that the new document was not published in the old collection, improving the precision of the results.
5. When a new document matches all the previous conditions, any duplicate of this document cannot be considered a new match. With this condition we eliminate duplicates among new documents.
6. When two old documents match all the previous conditions, any duplicate of one of these old documents cannot be considered as a new match. For example, consider that two old documents A and B are used to compose a new document. If later B and C are candidates to compose another new document and, if C is a duplicate of A, the new match is not considered. With this condition we eliminate duplicates among old documents.

Notice that with all these conditions we may incorrectly filter documents. For example, maybe a document with an old URL has a new content copied from old documents (see condition 4 above). Maybe a user really used queries to find documents to copy but the user copied only few shingle paragraphs (see condition 1). Maybe a user used only one document returned from a query to compose the new document (see conditions 2 and 3). We do not care about these situations. We are concerned in reducing as many as possible of the false matches, i. e., to avoid finding a new document that was not composed because a user performed a query.

4.2 Experimental Setup

In the experiments we used sets of the most frequent queries performed in a given period. We selected the most frequent queries because if more users performed that query, then it is more probable that one of the users has done it to compose a new document. We sorted the queries by their frequencies, eliminated the top 1,000 queries (they are many times navigational queries or related to pornography) and considered the following 15,000 top queries. For every query log these 15,000 queries represent approximately 14% of the user requisitions in the search engine.

Table 5 presents the meta data related to the three query logs used. In some experiments we used this log, as we present in Section 4.3.

Table 5. Characteristics of the logs.

Query log	Log period	Most freq.	Least freq.
2002	Aug/02 – Jun/03	640	71
2003	Sep/03 – Dec/03	168	23
2004	Feb/04 – Jan/05	449	51

The log periods are related to the period that the collections presented in Table 1 had been used as data in the search engine. We did not consider one month of log before and after each crawl, once we do not know exactly when

the new database was indexed and refreshed in the operating search engine. For example, the collection 2002 was crawled in July, 2002 and the collection 2003 was crawled in August, 2003. The query log 2002 considers the period between August, 2002 and June, 2003.

4.3 Experimental Results

This section presents the experimental results related to the log based content evolution study. The experiments are in general based on the following criteria: compare the number of documents returned by the algorithm presented in Section 4.1 (Figure 2) that obey all the six conditions shown in Section 4.2, using: (i) the real log for a given period, and (ii) a log of another period. For example, if we use the collection pair 2003–2004 as data set, in the situation *(i)* above we would use the query log 2003 shown in Table 5. This is the real query log for the period from 2003 to 2004. The query log 2002 (or 2004) could be used for the situation *(ii)* above, that is a query log of a period distinct of 2003–2004, the period which the collection was used in the search engine. To support our hypothesis, more documents must be returned for the situation *(i)* (using query log 2003), that simulate real query requisitions occurred between 2003 and 2004.

In general our collections have been crawled in a very distant period one from another. Table 1 in Section 2 presents the period of each crawl. From collection 2002 to 2003, there is an interval of 13 months, equivalently from collection 2004 to 2005. The period from collection 2003 to 2004 is the shortest: six months.

In order to choose the collection pair to be used in the experiments we observed the average lifespan of Web documents. The lifespan of a document is the difference between the date that it was deleted and the date that it was created [9]. Junghoo Cho [10] found that the average lifespan of Web documents is between 60 and 240 days, considering a sample of 720,000 documents from popular sites. Brewington et al. [11] found that the average lifespan is approximately 140 days for a data set of 800 million documents. Other works [12–14] present similar values, also considering other measures besides the average lifespan.

If we choose an old–new collection pair crawled 390 days longer one apart from another, it is likely that many new documents composed using old documents are no more detected in the new collection, due to the lifespan of the new document. For this reason we choose the collection pair 2003–2004 as old and new collections for the first experiment set.

Our first experiment set consists of the three frequent query logs presented in Table 5 for the collection pair 2003–2004, using the algorithm presented in Figure 2. Our hypothesis is that some users performed queries in collection 2003 for composing new documents, that were published in the collection 2004.

Figure 3 presents three curves for the query logs 2002, 2003 and 2004, from 5 to 30 minimal number of identical paragraphs (see Definition 5). For the query log 2003 the algorithm returned much more documents than for the other logs for any minimal number of identical paragraphs considered. It represents an evidence that people make searches to find a content and create a new document.

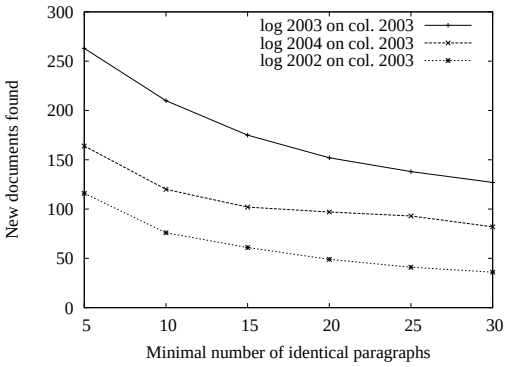


Fig. 3. Query logs 2002, 2003 and 2004 used for the collection pair 2003–2004 for different minimal number of identical paragraphs.

According to Figure 3, the use of the query logs 2002 and 2004 also returned some new documents. More documents are returned using the query log 2004 than the query log 2002. We highlight some possible reasons for these figures:

- It is possible that the query log 2003 has more similar queries to the query log 2004 than to the query log 2002.
- It is possible that queries that returned new documents with the query log 2004 were not in the set of the 15,000 frequent queries considered in the query log 2003, but occurred in another part of the query log 2003.
- It is possible that some documents returned with the query log 2004 (or also with the query log 2002) were composed by old documents returned in two or more different queries performed by the user in a session.
- It is possible that the old documents were returned together by other queries in another search engine.

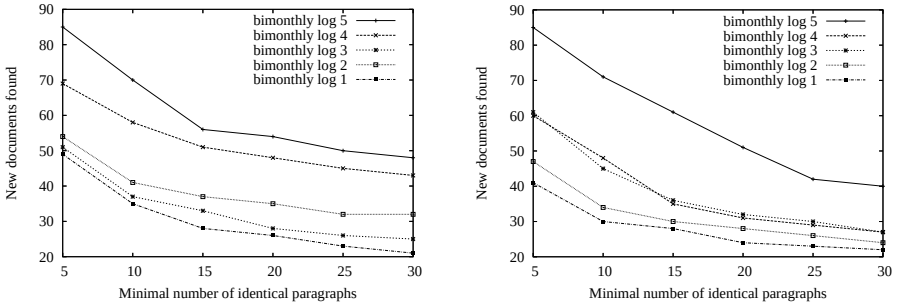
In the second experiment set we used parts of the logs shown in Table 5. We divided the query logs 2002 and 2004 into five bimonthly logs. For example, in log 2004 we considered the months February and March as being the bimonthly log 1, the months April and May as being the bimonthly log 2, and so on, until the months October and November as being the bimonthly log 5. We preferred not to use the remaining month in the log, December, since this log would have queries with about half of the frequency of the bimonthly logs, what probably would bias the results.

For each bimonthly log we sorted the queries by their frequencies, eliminated the top 1,000 queries and considered the 5,000 top queries. We used fewer queries than the previous logs (in which we used 15,000 queries) because now the period is shorter (two months) and we are not interested in less frequent queries. Table 6 presents information about the bimonthly logs. The average values considered in the table are related to the five bimonthly logs used for each year.

Figure 4 presents the number of documents returned when *a*) the five 2002 bimonthly logs are used for the collection pair 2002–2003, and *b*) the five 2004

Table 6. Characteristics of the bimonthly logs.

Query log	Log period	Average most freq.	Average least freq.
2002	Aug/02 – May/03	98	27
2004	Feb/04 – Nov/04	149	27



(a) Bimonthly logs from query log 2002, used for collection pair 2002–2003.

(b) Bimonthly logs from query log 2004, used for collection pair 2004–2005.

Fig. 4. Bimonthly logs study.

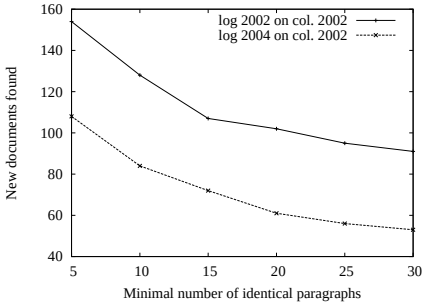
bimonthly logs are used for the collection pair 2004–2005. Bimonthly log 5 is the most recent bimonthly log and bimonthly log 1 is the oldest bimonthly log.

According to the figures, the most recent bimonthly logs returned more documents than older bimonthly logs. This would be expected, considering that many documents composed by documents returned by queries in the oldest bimonthly logs do not exist any more in the more recent collection of the pair, due to the lifespan of the documents.

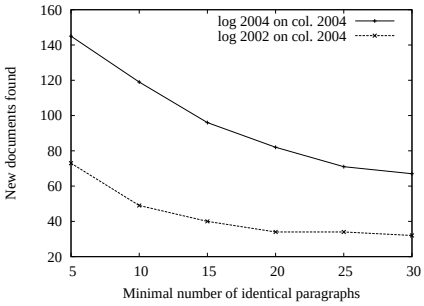
Considering that the average lifespan of a Web document is about 140 days [11], equivalently 4.5 months, the fact of finding a great number of documents for the two most recent bimonthly logs from both query logs 2002 and 2004 is another evidence that users performed queries in the search engine before composing their new pages with old content.

The third experiment set uses the bimonthly logs 4 and 5 (the most recent bimonthly logs) from both query logs 2002 and 2004 for both collection pairs 2002–2003 and 2004–2005. We expect better results running the bimonthly logs from 2002 for the collection pair 2002–2003 and the bimonthly logs from 2004 for the collection pair 2004–2005, since they are the real simulation of users performing queries in the past.

Figure 5 presents the number of documents returned when *a*) bimonthly logs 4 and 5 from 2002 and 2004 are used for collection pair 2002–2003 and *b*) bimonthly logs 4 and 5 from query logs 2002 and 2004 are used, for collection pair 2004–2005. When the real data is used (logs 2002 in pair 2002–2003 and logs 2004 in pair 2004–2005) the result is substantially better. The comparison of the two curves in each plot provides another piece of evidence towards the consistency of our hypothesis.



(a) Bimonthly logs (2002 and 2004) used for collection pair 2002–2003.



(b) Bimonthly logs (2002 and 2004) used for collection pair 2004–2005.

Fig. 5. Distinct bimonthly log sets used in the same collection.

As a conclusion, we have presented evidence for arguing that the stated hypothesis is valid, in various ways, considering our data set. We clearly discard the possibility that all the results found and shown in this section are just coincidences.

5 Web Content Evolution Study

In this section we study how old content is used to compose new documents. Section 5.1 presents our algorithm to find the parents and children. Section 5.2 presents the results for the Chilean Web, using distinct old–new collection pairs.

5.1 Algorithm Description

In this section we describe our algorithm to study the content evolution of the Web. Similarly to the algorithm presented in Section 4.1, this algorithm is composed of two stages. The objective of the first stage is to find new similar documents (see Definition 6). The objective of the second stage is to select parents from a candidate set.

The first stage of the algorithm consists of randomly selecting a sample of old documents (from an old collection O), inserting their shingles into a hash table, and searching for the shingles of each new document (from a new collection N) in the hash table. With exception of how the old documents are selected, this step is similar to the first step of the algorithm of the log-based content evolution study presented in Section 4.1.

Figure 6 presents the second stage of the algorithm. N_i is the new similar document, O_j is the correspondent old document with some similarity with N_i and $minNum$ is the minimal number of identical paragraphs.

The algorithm of Figure 6 initially filters the new similar documents with the minimal number of identical paragraphs equals to 10. This condition is applied to eliminate false matches, since we manually verified that many old–new document pairs with short overlap have only formatting in common, that was not cleaned by the crawler system.

```
1 For each document pair (N_i, O_j)
2    If minNum > 10
3       If it is the first time that O_j is a parent and O_j URL is found in new col.
4          Increment the number of equivalents;
5       Else
6          If it is the first time that N_i or a duplicate of N_i is a child
7             Increment the number of children;
8             If it is the first time that O_j is a parent
9                Increment the number of parents;
```

Fig. 6. The second stage of the algorithm to study content evolution of the Web.

The algorithm verifies if O_j is found in the new collection (step 3). If it is found, the number of equivalent documents (see Definition 8) is incremented. If it is not the first occurrence of O_j, it is not searched again.

After verifying if the documents are equivalent, the algorithm verifies if N_i is a child of O_j. The condition represented in step 6 of the algorithm is a way of eliminating duplicates in the new collection. Consider that a document A has thousands of duplicates in both collections old and new. It is probable that if we randomly choose about 5% of the old collection, one of the duplicates of A will be chosen. If we allow duplicates in the new collection, every duplicate of A in the new collection will be considered as a child, introducing noise in the results.

Finally, if the condition of the step 6 is true, N_i is a child of O_j. The child is classified and the number of parents is incremented, what happen only if N_i is the first child of O_j.

5.2 Chilean Web Content Evolution

We study the content evolution for the Chilean Web by randomly choosing documents from collections 2002, 2003 and 2004, and observing the occurrence of parts of these documents in the most recent collections. Table 7 presents the number of parents in collection 2002 that generate children, grandchildren and great-grandchildren, respectively in collections 2003, 2004 and 2005. The random sample contains 120,000 documents from collection 2002.

Table 7. Number of equivalent documents and parents in collection 2002 that generated descendants.

collection pairs	2002–2003	2002–2004	2002–2005
# of parents	5,900	4,900	4,300
# of children	13,500	8,900	9,700
# of equivalents	13,900	10,700	6,800

According to Table 7, 5,900 documents of the collection 2002 are parents of 13,500 documents in the collection 2003 for the sample considered. We see that 8,900 documents in the collection 2004 are grandchildren of documents in the collection 2002, and that 9,700 documents in the collection 2005 are great-grandchildren of documents in the collection 2002.

Table 8, in turn, presents the number of parents in collection 2003 that generate children and grandchildren, respectively in collections 2004 and 2005 (for

a random sample of 120,000 documents from collection 2003). In relation to the collection 2003, 5,300 documents are parents of documents in the collection 2004 and 5,000 are grandparents of documents in the collection 2005. The sample considered in collection 2003 generated content in 33,200 documents of the collection 2004 and 29,100 documents of the collection 2005.

Table 8. Number of equivalent documents and parents in collection 2003 that generated descendants.

collection pairs	2003–2004	2003–2005
# of parents	5,300	5,000
# of children	33,200	29,100
# of equivalents	19,300	10,500

The collection 2003 generated many more children than collection 2002. We suppose this is due to the fact that the Chilean Web of 2002 was not crawled in a large part to create collection 2002 (see Table 1). Thus, many documents in the most recent collections were composed by documents existent on the Web in 2002 but not existent in the collection 2002.

Observing Tables 7 and 8 we see that the number of children is always considerably greater than the number of parents. For the collection pair 2003–2004 there are, on average, more than six children for each parent. Thus, few documents are copied many times, so the number of documents that do not generate a child is smaller than the number of documents that do not have parents.

Now we observe the evolution of the number of children and the number of equivalent documents in these years. From collection pair 2003–2004 to collection pair 2003–2005 the number of children reduced only 12.5%, while the number of equivalent documents reduced 45.5%. From collection pair 2002–2004 to collection pair 2002–2005 the number of children increased.

We conclude that the number of copies from previously copied documents is indeed greater than the number of documents copied from random old documents. An open question is: do the search engines contribute to this situation, since they privilege popular documents [15, 16] and people use search engine to compose new documents? (according to the evidences previously presented in this paper). We recently finished a deeper study that gives strong evidence that this hypothesis is true [17].

6 Related Work

In this section we present works related to finding and eliminating duplicates and near-duplicates on the Web, and works related to the dynamics of the Web content.

Broder et al. [1] used shingle to estimate the text similarity among 30 million documents retrieved from a walk of the Web. The similarity was evaluated using a sample of fixed size (a fingerprint) for each document. Considering a resemblance

of 50%, they found 2.1 million clusters of similar documents, a total of 12.3 million documents.

Shivakumar and Garcia-Molina [6] crawled 24 million Web documents to compute the overlap between each pair of documents. Pieces of the documents are hashed down to a 32-bits fingerprint and stored into a file. A similarity is detected if two documents share a minimal number of fingerprints. The number of replicas is estimated as approximately 27%. Cho, Shivakumar and Garcia-Molina [5] combined different heuristics to find replicated Web collections. They used 25 million Web documents and found approximately 25% of duplicates.

Fetterly, Manasse and Najork [7] extended the work by Broder et al. [1] in terms of the number of compared documents and investigated how clusters of near-duplicate documents evolve with the time. They found that clusters of near-duplicate documents are fairly stable and estimated the duplicates as approximately 22%.

Ntoulas, Cho and Olston [12] crawled all pages from 154 sites on a weekly basis, for a period of one year, studying some aspects of the Web evolution, such as birth, death, and replacement of documents. They found that every week 8% of the pages are replaced and about 25% are new created links. With respect to the pages that do not disappear over time, about 50% do not change at all, even after one year. Additionally, those that do change, only undergo minor changes in their content, and even after a whole year 50% of the changed pages are less than 5% different from their initial version. In a similar work using the same data set, Ntoulas et al. [18] found that after a year, about 60% of the documents and 80% of the links on the Web are replaced.

Cho and Roy [15] studied the impact of search engines on the popularity evolution of Web documents. Given that search engines currently return popular documents at the top of search results, they showed that newly created documents are penalized because these documents are not very well known yet. Baeza-Yates, Castillo and Saint-Jean [16] showed that Pagerank [19] is biased against new documents, besides obtaining information on how recency is related with the Web structure. This fact supports our findings, given that we show that ranking algorithms are biasing the content of the Web. From the perspective of a search engine user, the Web does not evolve too much, considering that the new content is partially composed by the content of old popular documents.

Mitzenmacher [20] introduced a dynamic generative user model to explain the behavior of file size distributions (not only Web text documents). He showed that files that are copied or modified are more likely to be copied or modified subsequently.

Our work differs from the above mentioned papers in four main aspects: i) we study duplicate documents in collections where all sites under a given domain (.cl, from Chile) were crawled, which represents accurate and representative subsets of the Web; ii) we compare the number of duplicates for the complete collection and for the reachable and unreachable components; iii) we associate the search engine ranking algorithms with the Web content evolution; and iv) we study how old content is used to create new content in new documents.

7 Concluding Remarks

In this paper we have presented a study about duplicates and the evolution of the Web content. We have shown that the Web has many more duplicates than previously acknowledged in the literature. Other works use collections crawled by following links on the Web. The number of duplicates found for the unreachable component is on average 74.6% higher than the number of duplicates for the reachable component. Once we have used accurate and representative subsets of the Web, we believe that our conclusions can be extended to other Web collections.

We have shown that a significant portion of the Web content has evolved from old content. We have also shown that this portion is partly biased by the ranking algorithm of Web search engines, as people use a query to select several sources to apply a cut and paste to create part or all the content of a new page.

Additionally, we have demonstrated that the number of copies from previously copied Web pages is indeed greater than the number of copies from other pages. An open question is: do the search engines contribute to this situation, since they privilege popular documents and people use search engine to compose new documents? If the answer is true, then search engines contribute to slow down the evolution of the Web.

As future work it would be interesting to study the characteristics of the documents in the unreachable component of the Web (most of the times they are more recent documents [3]). Maybe it is heuristically possible to separate the interesting new documents from other documents that are many times replications of documents in the reachable component of the Web graph.

Acknowledgements

This work was partially funded by Spanish Education Ministry grant TIN2006-15536-C02-01 (R. Baeza-Yates and A. Pereira) and by Brazilian GERINDO Project–grant MCT/CNPq/CT-INFO 552.087/02-5 (N. Ziviani and A. Pereira), and CNPq Grants 30.5237/02-0 (N. Ziviani) and 14.1636/2004-1 (A. Pereira). We also would like to thank Karen Whitehouse for the English revision.

References

1. Broder, A., Glassman, S., Manasse, M., Zweig, G.: Syntactic clustering of the Web. In: Sixth International World Wide Web Conference. (1997) 391–404
2. Broder, A.: On the resemblance and containment of documents. In: Compression and Complexity of Sequences (SEQUENCES'97). (1998) 21–29
3. Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. In: Ninth International World Wide Web Conference (WWW'00), Amsterdam, Netherlands (May 2000) 309–320
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to algorithms. MIT Press/McGraw-Hill, San Francisco, CA (1990)

5. Cho, J., Shivakumar, N., Garcia-Molina, H.: Finding replicated Web collections. In: ACM International Conference on Management of Data (SIGMOD). (May 2000) 355–366
6. Shivakumar, N., Garcia-Molina, H.: Finding near-replicas of documents on the Web. In: International Workshop on the World Wide Web and Databases (WebDB'98), Lecture Notes in Computer Science (1998) 204–212
7. Fetterly, D., Manasse, M., Najork, M.: On the evolution of clusters of near-duplicate Web pages. In: First Latin American Web Congress, Santiago, Chile (November 2003) 37–45
8. Calado, P.: The WBR-99 collection: Data-structures and file formats. Technical report, Department of Computer Science, Federal University of Minas Gerais (1999) http://www.linguateca.pt/Repositorio/WBR-99/wbr99.pdf.
9. Castillo, C.: Effective Web Crawler. PhD thesis, Chile University (2004) Chapter 2.
10. Cho, J.: The evolution of the web and implications for an incremental crawler. In: 26th Intl. Conference on Very Large Databases (VLDB), Cairo, Egypt (September 2000) 527–534
11. Brewington, B., Cybenko, G., Stata, R., Bharat, K., Maghoul, F.: How dynamic is the web? In: Ninth Conference on World Wide Web, Amsterdam, Netherlands (May 2000) 257–276
12. Ntoulas, A., Cho, J., Olston, C.: What's new on the Web? the evolution of the Web from a search engine perspective. In: World Wide Web Conference (WWW'04), New York, USA (May 2004) 1–12
13. Douglis, F., Feldmann, A., Krishnamurthy, B., Mogul, J.C.: Rate of change and other metrics: a live study of the world wide Web. In: Symposium on Internet Technologies and Systems USENIX, Monterey, CA (December 1997) 147–158
14. Chen, X., Mohapatra, P.: Lifetime behaviour and its impact on Web caching. In: IEEE Workshop on Internet Applications (WIAPP'99), San Jose, CA (July 1999) 54–61
15. Cho, J., Roy, S.: Impact of search engine on page popularity. In: World Wide Web Conference (WWW'04), New York, USA (May 2004) 20–29
16. Baeza-Yates, R., Castillo, C., Saint-Jean, F.: Web dynamics, structure and page quality. In Levene, M., Poulovassilis, A., eds.: Web Dynamics. Springer (2004) 93–109
17. Baeza-Yates, R., Pereira, A., Ziviani, N.: Genealogical trees on the web: A search engine user perspective. Submitted (2007)
18. Ntoulas, A., Cho, J., Cho, H.K., Cho, H., Cho, Y.J.: A study on the evolution of the Web. In: US – Korea Conference on Science, Technology, and Entrepreneurship (UKC), Irvine, USA (2005) 1–6
19. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the Web. Technical Report CA 93106, Stanford Digital Library Technologies Project, Stanford, Santa Barbara (January 1998)
20. Mitzenmacher, M.: Dynamic models for file sizes and double pareto distributions. Internet Mathematics **1**(3) (2003) 305–333