Genealogical Trees on the Web: A Search Engine User Perspective

Ricardo Baeza-Yates
Yahoo! Research
Ocata 1
Barcelona, Spain
rbaeza@acm.org

Álvaro Pereira*
Federal Univ. of Minas Gerais
Dept. of Computer Science &
Barcelona Media
Ocata 1, Barcelona, Spain
alvaro@dcc.ufmg.br

Nivio Ziviani
Federal Univ. of Minas Gerais
Dept. of Computer Science
Av. Antonio Carlos 6627, ICEx
Belo Horizonte, Brazil
nivio@dcc.ufmg.br

ABSTRACT

This paper presents an extensive study about the evolution of textual content on the Web, which shows how some new pages are created from scratch while others are created using already existing content. We show that a significant fraction of the Web is a byproduct of the latter case. We introduce the concept of Web genealogical tree, in which every page in a Web snapshot is classified into a component. We study in detail these components, characterizing the copies and identifying the relation between a source of content and a search engine, by comparing page relevance measures, documents returned by real queries performed in the past, and click-through data. We observe that sources of copies are more frequently returned by queries and more clicked than other documents.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation

Keywords

Web, text, content evolution, search engine, Web mining

1. INTRODUCTION

The Web allows everybody the opportunity to become a publisher. Entities like companies, products, services, and people can be represented on the Web. One supposes that many of these potential publishers either have insufficient content or do not know how to represent their interests. Hence, some of the publishers refer to the Web itself to find good representations for their entities.

Little is known about the evolution of the textual content on the Web. We know how Web components (such as URLs and figures) evolve [11] and how the structure evolves [3], but not how the content evolves. Our work provides the first step towards understanding how old content is used to create new content. That is, we want to find the original sources, if any, of the content or part of the content of a new page. We regard each source as a *parent* of a new page, in order to define a *genealogical tree* for the Web. The study of the genealogical tree allows us to understand what portion of the pages are either totally new parents or parents that are children of other parents.

Our experiments consider several representative snapshots of the Chilean Web and one snapshot of the Spanish Web. We estimate that 23.7% of new Web documents that appeared within a span of a year have content from previously published documents (see Section 6.5 for estimations). Most of them represent inter-site copies (approximately 75%), in which the publishers use content from a parent document from another site, and they need to find this document.

Web search engines are widely used to provide users with content that approximates what they are looking for. Web publishers are also Web users, and frequently are advanced search engine users. It is natural that if they need to find content on the Web, they perform a query on a search engine.

In this direction, in addition to the genealogical tree study, we analyze whether there is any association between the sources of reused content (the parents) and the results of real queries from a search engine log. We see that parents are more connected to the Web graph and have a much higher Pagerank than other pages. Probably as a consequence, parents appear more often as results of queries and are much more clicked, which is shown in our analysis.

Our results are evidence that some Web publishers actually performed queries in order to find some content and republish. Thus, the conclusion is that part of the Web content is biased by the ranking function of search engines. Exploring our results beyond the scope of this paper would explain the impact of the user's copy behavior on the quality of the search engine results, and how search engine designers can profit from that behavior, for example by associating a better page quality value for a previously low-quality page that is used as source of copy. In this case a child page would inherit properties of its parent (in case they are not duplicates or near-duplicates, that is, only part of the content is copied).

The main contributions of this paper are: (i) to propose a methodology to study the genealogy of the Web content; (ii) to study the evolution of textual content on the Web, *i.e.*, how pieces of documents are reused; (iii) to generalize the content reuse results to the whole Web (or other subsets of the Web), providing an estimation of how much content is reused on the Web; and (iv) to study how search engine ranking algorithms may influence the evolution of Web content. To the best of our knowledge, these contributions are not covered in previous works.

*This work was done when at Yahoo! Research Barcelona as a long-term Ph.D. intern.

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

2. RELATED WORK

In this section we present some related work, starting with a couple of works that study the evolution of Web pages, and finishing with works about Web archiving. Despite the importance of these papers as related work, they have different approaches and different objectives, in comparison to our paper.

Ntoulas, Cho and Olston [12] studied some aspects of the Web evolution, such as birth, death, and replacement of documents. They crawled all pages from 154 sites on a weekly basis, for a period of one year. In a similar work using the same data set, Ntoulas et. al. [11] found that after a year, about 60% of the documents and 80% of the links on the Web are replaced.

Cho and Roy [8] 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 well known yet. Pandey *et al.* [14] proposed a simple solution to this problem, based on the introduction of a controlled amount of randomness into search result. Baeza-Yates *et al.* [2] showed that PageRank [13] is biased against new documents.

On the other hand, Fortunato *et al.* [10] showed that popular sites receive far less traffic than predicted, suggesting that the possible bias introduced by search engines does not lead to monopoly of information.

In a recent work, Toyoda and Kitsuregawa [16] proposed the notion of a "novelty measure" to estimate if a new linked URL is really new or if it is old but was not crawled for previous snapshots. The novelty measure is applied to an archive search engine, where new pages can be identified. Zhang and Suel [17] proposed a general framework for indexing and query processing of archival collections. By storing the documents in parts, and considering that in archiving a great portion of the data is replicated, their approach results in significant reductions in the index size and query processing cost.

3. CONCEPTUAL FRAMEWORK

3.1 Document Representation

We use the concept of shingles [6] to represent a document (the document fingerprint). A **shingle paragraph** (also referred in this paper as just "shingle" or "paragraph") is a sequence of three sentences of the document. A **sentence** is a sequence of words ended by a period. 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).

Each document is represented by the list of its shingles paragraphs, with overlap of sentences. As an example, suppose we have a document D_1 containing seven sentences: $D_1 = s_1.\ s_2.\ s_3.\ s_4.\ s_5.\ s_6.\ s_7$, where s_i, $1 \leq i \leq 7$, is a sentence of the text, and i is the order of occurrence of the sentences in the text. The shingle paragraphs for D_1 are: "$s_1.\ s_2.\ s_3.$", "$s_2.\ s_3.\ s_4.$", "$s_3.\ s_4.\ s_5.$", "$s_4.\ s_5.\ s_6.$", "$s_5.\ s_6.\ s_7.$".

In our experiments we considered only documents with more than 450 characters and at least three shingle paragraphs, or equivalently five sentences. Preliminary experiments demonstrated that for considering two documents similar, it is necessary to have a minimal percentage of similarity between them, trying not to find many false matches (occurring in cases that only one or two popular shingle paragraphs are identical). We did not consider short documents because they cannot be represented by a minimal number of shingle paragraphs, and thus cannot be compared with others. Around 25% of the documents in each collection were removed from consideration for these reasons.

3.2 Document Instance

We define a **cluster** as a set of documents with exactly the same textual content for a given collection. Each document in a collection is either (i) **duplicate**, if it belongs to a cluster, or (ii) **unique**, otherwise.

Each different content in a given collection is represented as different **instances**. If a set of documents are duplicates among them, their contents are the same and they are represented by a unique instance. If a document is unique, its content is represented by an instance.

Most of the studies and conclusions presented in this paper are concerned with the instance rather than with the document. The collections have a large number of duplicates, and thus it is wrong to say that every duplicate in the same cluster is a parent when part of the duplicate's content is found in a more recent collection. The concept of instance represents an important solution for the duplication problem in this work, since it compares content over different data sets.

3.3 Inter-Collection Relations

Consider a collection as being a snapshot of a given Web subset. Two documents, in two distinct collections, are **coexistent** (or they coexist), if their URLs are exactly the same. In this case, the same document URL exists in both collections (the content may differ). Two instances I_1 and I_2 in two distinct Web collections coexist, if at least one of the documents that I_1 represents has the same URL as one of the documents that I_2 represents.

An instance in a new collection has a **parent** instance in an old collection if it shares a minimal percentage of shingle paragraphs with the parent and the instance in the new collection is not represented in the old collection (it does not have a coexistent instance). The instance in the new collection is referred to as a **child**. The minimal percentage of shingle paragraphs used in this work is 20% (parent and child instances must share at least 20% of their content). After a manual analysis in a sample, we did not find false matches for this minimal similarity percentage.

A new instance is **orphan** if it does not have a coexistent instance or a parent in the old collection. An old instance is **sterile** if it does not have a coexistent instance or a child in the new collection.

In this paper we study two kinds of relations: **inter-site** and **intra-site**. Excluding the `http://` prefix from the URL, the remaining of the string before finding a slash gives the site to which the document belongs. Inter-site relations require that the parent and the child belong to different sites, whereas for intra-site relations the parent and the child belong to the same site. Our study treats these relations separately because intra-site relations tend to occur when publishers reuse the content of their own site. For inter-site relations the way in which the publishers find the parent is much more difficult to guess.

Mirrors were detected for inter-site relations (detection is not required for intra-site relations). Two sites are considered mirrors one of the other if at least 75% of their documents are clustered together (are duplicates in the same cluster) [5], and each site has at least 10 documents. This threshold guarantees that a minimal number of documents are clustered together.

3.4 Genealogical Trees on the Web

A genealogical tree on the Web is a representation for parents and children in different snapshots of a given Web subset. Each instance is classified into a different genealogical tree component.

For the description of the components, let P_t be the set of parents in a snapshot t whose children belong to a snapshot $t+1$. Let C_t be the set of children in a snapshot t whose parents belong to a snapshot $t-1$. *Each document of each collection* is labeled as one of the following components:

1. Without Relation: represents instances that are not parent or child instances in a collection. They are sterile and/or orphan instances.

2. Original Parents (OrP): represent parents that are not children neither were parents in the previous collection (generating some child in the current collection). This component represents parents that have no relation with the older collection. The original parents set in a collection is the difference between the parents set and the union of the children set and the parents set in the previous snapshot, as shown in Equation 1.

$$\mathrm{OrP}_t = P_t \setminus (C_t \cup P_{t-1}) \quad (1)$$

Notice that as we are looking for the original parent instances in snapshot t, P_{t-1} represents the parents in snapshot $t-1$ that still exist in snapshot t. We do not include coexistent instances in Equation 1 because it is obvious that for a parent in snapshot $t-1$ being a parent again in snapshot t, it has to exist.

3. Old Parents (OlP): represent instances that were parents in the previous collection, and are parents again in this collection. It means that they have some child in the current collection. The set operation shown in Equation 2 indicates how old parents are found.

$$\mathrm{OlP}_t = P_t \cap P_{t-1} \quad (2)$$

4. Children and Parents (CnP): represent instances that are children (with respect to the older collection) and parents (with respect to the newer collection), as shown in Equation 3.

$$\mathrm{CnP}_t = P_t \cap C_t \quad (3)$$

5. Sterile Children (StC): represent children that are not parents, as shown in Equation 4. This component represents children that have no relation with the more recent collection.

$$\mathrm{StC}_t = C_t \setminus P_t \quad (4)$$

For a given collection, each parent is classified as either original parent, old parent, or child and parent. It is easy to verify that $P_t = \mathrm{OrP}_t \cup \mathrm{OlP}_t \cup \mathrm{CnP}_t$. Equivalently, each children is classified as either child and parent or sterile child ($C_t = \mathrm{CnP}_t \cup \mathrm{StC}_t$). By definition, children and parents instances belong to both, the parents set and the children set.

Figure 1 illustrates a genealogical tree and its components. Every collection represented in this example has 10 instances. Continuous arrows represent parent/child relations and dashed arrows represent coexistent instances.

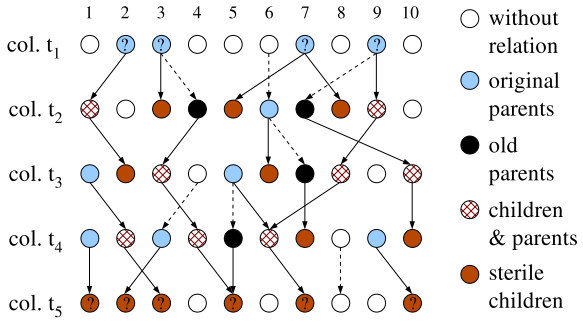

Figure 1: Example of the genealogical tree and its components.

Notice that for the oldest collection of the data set, represented as $col.t_1$, it is not possible to classify a parent because there is no data about parents of instances in this collection. In this case we represent all the parents as original parents, but we know that a portion of them must be in a different class. Equivalently, for collection t_5, it is not possible to know which instances are children and parents or which ones are sterile children. These documents are represented in the figure with a question mark.

4. ALGORITHMS SUMMARY

In this section we summarize the main algorithms designed and implemented for this work. Basically we present the algorithm to detect duplicates, the algorithm to find parent and child document candidates, the algorithms to filter the candidates in order to return parent and child instances, and the algorithm to select the parents, in order to associate only one parent for each child. Although separately the algorithms are not new and have no innovative aspects, their combination for the purpose of analyzing the Web content evolution is new and has successfully been employed.

4.1 Duplicate Detection

The algorithm to find duplicates works by clustering documents with the same content [9]. Initially, collection C (with n documents) is divided into m sub-collections S_i, $0 \leq i < m$. The algorithm runs in m steps. For each sub-collection S_i, $0 \leq i < m$, the text of the documents in S_i is first inserted into a hash table.

Next, the documents of C are searched in the hash table. For each new duplicate pair found, a new cluster is created and the duplicate pair is inserted into the new cluster. 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 of each iteration i, the sub-collection S_i is excluded from C ($C = C - S_i$). At the end, the algorithm returns a set of clusters, with each cluster containing a list of duplicate documents.

4.2 Detecting Candidates

The algorithm to detect candidate parents and children is similar to the algorithm to detect duplicates, summarized in the previous section. The main differences are the number of collections involved and the representation of the document (now the shingles are used to represent the document, as described in Section 3.1). Instead of searching for documents of the same collection, the algorithm to find parents and children is applied to a pair of old–new Web collections.

The shingle paragraphs of the old collection are inserted into the hash table (in parts) and the shingle paragraphs of the new collection are searched. If a new document shares three or more shingle paragraphs with some document of the old collection, the old-new document pair is stored as candidate. At the end, for each old document, a list of child candidates is stored.

4.3 Finding Parent and Child Instances

After finding parent and child document candidates, two steps are now required: obtaining the list of parent and child instance candidates, and filtering the parent and child instances from the candidates.

Figure 2 summarizes the algorithm to obtain parent and child instance candidates. Along the first loop the old documents are instantiated, and along the second loop new documents found as child instance candidates, are instantiated. With this second loop, the list of child candidate documents for each old instance is used to generate the list of child instance candidates for each old instance.

```
For each old document ODi
    If ODi is unique
        Create an old instance OIk;
        Keep the list of child candidates Cj of ODi to OIk;
    Else
        If it is the first occurrence of the ODi cluster
            Create an old inst. OIk assoc. to the ODi cluster;
            Keep the list of child cand. Cj of ODi to OIk;
For each old instance OIk
    For each child candidate Cj in the list of OIk
        If Cj is unique
            Make Cj be a child candidate instance CIn;
            Include CIn in the list of child inst. cand. for OIk;
        Else
            If it is the first occur. of Cj cluster as cand. for OIk
                Make the Cj cluster be a child cand. inst. CIn;
                Include CIn in the list of child inst. cand. for OIk;
```

Figure 2: Algorithm to obtain parent and child instance candidates in a collection pair.

Figure 3 summarizes the algorithm to filter candidate instances and find parents and children for a collection pair. The algorithm works by labeling old and new found instances as parent-child instances or as coexistent instances. If both documents of a parent-child candidate pair are labeled as coexistent, this pair cannot be a parent-child, although other child candidate in the list of the parent candidate can become a real child. In this case, the old instance is labeled as parent and coexistent, meaning that the parent exists in the new collection but a new document was generated with its content in the mean time.

```
For each old instance OIk
    For each child candidate instance CIn
        For each ODi ∈ OIk
            For each Cj ∈ CIn
                If URL(ODi) = URL(Cj)
                    Label OIk and CIn as coexistent;
        If CIn is not a coexistent
            If OIk and CIn share at least 20% (threshold) parag.
                Label OIk as parent and CIn as child, associating them;
For each old instance OIk
    Classify it as either coexistent, parent, par. and coex., or sterile;
For each new instance
    Classify it as either coexistent, child or orphan;
```

Figure 3: Algorithm to filter candidates to find instances of parents and children.

4.4 Selecting Parents

The output of the algorithm presented in Figure 3 can be used to classify each document into a different component of the Web genealogical tree. For our specific study, we follow processing the data in order to associate only one parent for each child. This association is required because every near-duplicate instance in the old collection is considered a parent when one of the near-duplicates has a child. We ran preliminary experiments and detected a high number of parents. They expressively introduced noise to the results, impeding the correct classification of parents.

If we detected near-duplicates instead of duplicates (see Section 4.1), we could have inaccurate results. First, because, clusters of near-duplicates are intrinsically not accurate. Supposes that a page A shares 70% of content with a page B, which shares 70% with C. It is possible that A and C share only 40% of their content, making the decision of which documents to cluster together a hard task. Second, because we study the evolution of content reuse in small parts of documents. The minimal similarity allowed is 20%, which is too low to perform clusters of near-duplicates.

For each child instance, we select a parent instance from its list of parents. The parent that shares the highest number of paragraphs with the child is selected. When the number of paragraphs is the same for more than one parent (this situation is not frequent), it does not mind which parent is chosen. In this case we select the parent with the lowest identifier. This heuristic is used just in order to select the same document in case of this list occur again for another child.

After associating a parent for each child, we separate intra-site and inter-site relations, and apply the mirror filter for inter-site relations.

5. DATA SET

For the experiments we have used five collections of pages of the Chilean Web, crawled in five distinct periods of time, from July 2002 to February 2006. Table 1 presents the main characteristics of the collections. The HTML tags were excluded from the documents, thus the metadata in the table represents data on the text found in the documents in each collection.

Table 1: Characteristic of the collections.

Col. name	Crawling date	total number of docs. (mi)	Size (Gbytes)
2002	Jul 2002	1.04	2.3
2003	Aug 2003	3.11	9.4
2004	Jan 2004	3.13	11.2
2005	Feb 2005	3.14	11.3
2006	Feb 2006	3.72	14.5

Each collection was crawled by the Chilean search engine TodoCL [15]. In order to compose the collections, the complete list of the Chilean Web primary domains were used to start the crawling, guaranteeing that a set of pages under every Chilean domain (`.cl`) was crawled, once the crawls were pruned by depth. Domains outside the Chilean primary domain were only crawled if their IP address was from a Chilean IP provider. The collections have sucessfully been used for other works in characterizing the Web [3].

Any Web collection is a biased and partial image of the Web [4]. We decided to use the Chilean collections because the way in which they were crawled indicates that they are the least biased data set for the kind of study we have done. As far as we know, Chile is the only country where a series of annual snapshots have been collected, using as seed the complete list of Chilean domains.

6. GENEALOGICAL TREE

In this section we present our study of the Chilean Web genealogical tree. Most of the results are presented as percentages in relation to instances. The number of instances for collections 2002, 2003, 2004, 2005 and 2006 is 416,300, 1,262,900, 1,157,100, 1,396,200, and 1,808,500, respectively.

6.1 Coexistent Instances

Figure 4 presents the percentage of **coexistent** instances among each collection pair, in relation to the old collection (first bar of each pair) and in relation to the new collection (second bar of each pair).

For instance, around 41% of the instances in the old collection 2002 continue to exist in collection 2003. These instances represent

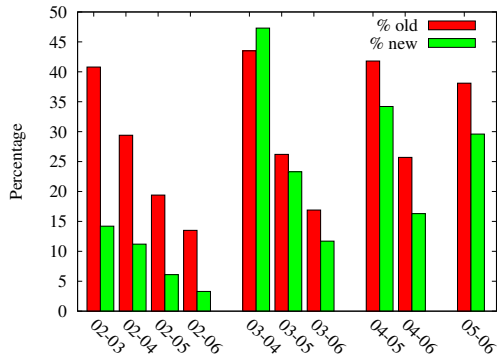

Figure 4: Percentage of coexistent instances among collection pairs.

around 14% of the new collection, 2003. The percentage changes due to the difference of the number of instances in each collection. Notice that the percentage of coexistent instances among the years (fixing the old collection) decreases linearly according to the time. The exception is collection pair 2003-2004, in which the difference of time from collection 2003 to 2004 is short, resulting in a large number of coexistent instances.

6.2 Parents and Children

In this section we study the number of parents and the number of children for collection pairs. Table 2 presents the number of parent instances, child instances and child documents, for both intra-site and inter-site relations. The number of child documents is calculated by counting the number of documents represented by a child instance, that is, the number of documents in a cluster of an instance.

Table 2: Number of parent instances, child instances, and child documents, for intra-site and inter-site relations, for each collection pair (in thousands).

	Intra-site			Inter-site		
col.	par.	child	ch. doc.	par.	child	ch. doc.
02-03	13.7	23.7	41.5	12.7	27.0	63.8
02-04	10.1	12.1	18.6	11.4	39.2	67.8
02-05	10.3	12.4	21.3	10.1	32.1	44.5
02-06	8.6	10.5	14.2	9.5	20.0	38.6
03-04	21.3	41.3	69.5	19.0	70.9	115.1
03-05	29.9	39.9	54.8	21.7	65.1	99.5
03-06	26.5	31.7	40.8	22.8	60.8	126.8
04-05	34.6	43.6	62.4	20.1	51.9	83.5
04-06	29.8	34.0	46.5	28.2	64.5	132.6
05-06	27.7	40.3	58.2	23.7	64.0	144.2

For instance, 12,700 inter-site parent instances in collection 2002 generated 27,000 instances is collection 2003. These 27,000 instances represent a total of 63,800 documents in collection 2003. Data show that, on average, the number of intra-site parents is higher than the number of inter-site parents. On the other hand, the number of intra-site children is much lower than the number of inter-site children. On average, an intra-site parent generates 1.37 child instances and 2.07 child documents, whereas an inter-site parent generates 2.78 child instances and 5.05 child documents.

Comparing the coexistent data presented in Section 6.1 and relation data presented in this section, we see that the percentage of coexistent instances decreases according to the time more than the number of parents and children decreases. For example, from collection pair 2002-2003 to collection pair 2002-2006, the number of coexistent instances in relation to the old collection decreases 70%, whereas the number of parents decreases 25% and the number of children decreases 26%, for inter-site relations. Furthermore, the number of inter-site children increases in some cases. For instance, collection pair 2002-2003 has 27,000 children, whereas collection pair 2002-2004 has 39,200 children.

The values presented above indicates that from 2002 to 2006 (and also from 2003 to 2005), many pages died and could not generate a child, but the children of part of those died pages became parents, generating new children and propagating the content. In this case, for example in collection 2002, part of the parents of documents in collection 2004 or 2005 are in fact grandparents. In next sections we present further discussions about this behavior.

Figure 5 plots the percentage of parent instances, relative to the number of instances of the old collection; the percentage of child instances, relative to the number of new instances for the new collection; and the percentage of child documents, relative to the number of new documents for the new collection. Both intra-site and inter-site relations are presented. We focus our study on the adjacent collection pairs, that is, 2002-2003, 2003-2004, 2004-2005 and 2005-2006. The percentage of children is lower than the percentage of parents only for collection pair 2002-2003, because collection 2002 is considerably smaller than collection 2003.

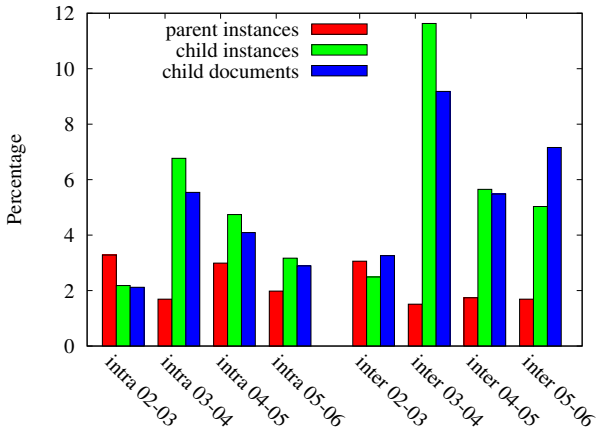

Figure 5: Percentage of parent instances, child instances, and child documents, for intra-site and inter-site relations.

Collection pair 2003-2004 presents the highest number of children for both intra-site and inter-site relations. This collection pair represents the shortest elapsed time between the crawling periods, as shown in Table 1, suggesting that relations are easier to be identified in closer periods (the difference of one year among the collections crawls may be enough for part of the children die).

It is important to notice that the percentage of inter-site children is considerably higher than the percentage of intra-site children, and that the sum of both percentages represent the total percentage of children. On average, 4.5% of old instances are parents, 10.4% of new instances are children, and 9.9% of new documents are children.

6.3 Linkage Among Relatives

In this section we study how children acknowledge their parents with links to them. The collections have no external links, so that

we are not able to study links to parents that no longer exist in a collection.

Table 3 presents the number of parents divided by the number of parents acknowledged by a child, for both intra-site and inter-site parents. For instance, one in each 8.4 intra-site parents in collection pair 2002-2003 are acknowledged by a child. The lower the value in the column "link", the higher the total number of links from children to parents for a collection pair.

Table 3: Number of parents divided by the number of parents acknowledged by a child.

Col. pair	intra-site	inter-site
02-03	8.4	35.3
02-04	8.8	39.8
02-05	7.1	66.6
02-06	14.1	257.3
03-04	9.5	147.8
03-05	8.0	142.4
03-06	27.8	366.2
04-05	17.0	119.2
04-06	50.5	479.7
05-06	15.1	209.3
average	16.6	186.4

Intra-site parents are much more acknowledged than inter-site parents. This is simple to understand, as internal links in a site are usually common. Furthermore, the number of acknowledged inter-site parents is also relatively high: one link for every 35.3 inter-site parents, as observed for collection pair 2002-2003, is a significant value, given that the probability of a parent being acknowledged by a random document is extremely low.

6.4 Chilean Web Genealogical Tree

In this section we present the components of the Web genealogical tree as defined in Section 3.4, for the Chilean Web data. Table 4 presents the percentage of parents for both intra-site and inter-site relations, considering only the intermediate collections, in which the genealogical tree components can be studied. The second column presents again, for comparison purposes, the number of parents. The following columns present, respectively, the percentage of original parents, old parents, and children and parents (see the definitions in Section 3.4).

Table 4: Percentage of parents for each component.

Collection	# parents (thousands)	original parents	old parents	child. & par.
intra 2003	21.3	96.5	0.7	2.8
intra 2004	34.6	87.1	2.1	10.8
intra 2005	27.7	87.5	2.3	10.3
inter 2003	19.0	90.0	2.1	7.9
inter 2004	20.1	81.1	7.9	11.0
inter 2005	23.7	88.1	3.6	8.3

According to the genealogical tree definition, if a given document exists in a time t_0 and generates a child in a time t_1, if the document is not crawled in t_1 and is crawled in a time t_2 (it skips snapshot t_1), it would wrongly be associated as a child of its own child (which would be considered a child and parent). For our data set, only 1.5% of the documents are skipped in a crawling, on average. In any case, we verified that they had negligible influence in our results, less than 5 documents wrongly appeared as children and parents.

Observing the 2003 collection in Table 4, we see that the percentage of original parents for this collection is the highest one among the three collections. This scenario is probably due to the small size of collection 2002. In this case many documents in collection 2003 should be children of documents in 2002, but their parents are not represented in collection 2002.

Data presented in the table demonstrate mainly two important issues. First, the percentage of children and parents and the percentage of old parents are higher for inter-site relations. Second, the percentage of children and parents is higher than the percentage of old parents. In order to understand these issues, Table 5 presents the probability of an instance becoming a parent in each component. The second column of the table presents the number of coexistent previous parents (referred as EPP), *i.e.*, the intersection between the parents set in snapshot $t-1$ and the coexistent instances set in snapshot t ($\mathrm{EPP}_t = P_{t-1} \cap E_t$, where E_t represents the coexistent instances between snapshots $t-1$ and t)

Table 5: Probability of an instance becoming a parent, for each parent component.

Data set	EPP (k)	$P(\mathrm{OlP})$	$P(\mathrm{CnP})$	$P(\mathrm{OrP})$
intra 2003	5.4	0.027	0.024	0.016
intra 2004	15.4	0.046	0.088	0.027
intra 2005	28.3	0.022	0.064	0.018
inter 2003	5.2	0.078	0.056	0.014
inter 2004	16.4	0.097	0.031	0.015
inter 2005	23.2	0.036	0.037	0.016

The number of coexistent previous parents, which is presented in thousands, is used to calculate the probability of a coexistent previous parent becoming a parent, presented in the third column of Table 5, where $P(\mathrm{OlP}_t) = |\mathrm{OlP}_t|/|\mathrm{EPP}_t|$ (see Section 3.4 for details about the variables). Given that an instance is coexistent and was a parent in the previous collection, the values in this column represent the probability of this instance becoming a parent.

The fourth column presents the probability of a child becoming a parent, given by $P(\mathrm{CnP}_t) = |\mathrm{CnP}_t|/|C_t|$. Note that data in this column represents the percentage of parent and child instances in relation to the number of sterile children. For instance, for inter-site relations in collection 2003, 5.6% of the children are classified as children and parents, whereas 94.4% are sterile children.

The fifth column presents the probability of an orphan instance becoming a parent (see Section 3.3 for the definitions), given by $P(\mathrm{OrP}_t) = |\mathrm{OrP}_t|/(|\mathrm{INS}_t| - (|\mathrm{EPP}_t| + |C_t|))$, where INS_t is the set of instances for snapshot t.

Table 5 shows that the probability of a child or a coexistent previous parent becoming a parent is higher than the probability of an orphan instance becoming a parent. This conclusion is true for both intra-site and inter-site parents, although for inter-site parents this probability is, on average, more than twice higher than for intra-site parents. In summary, an important conclusion is that instances with a previous relation (as either parent or child) are more likely to be parents than documents without relations. Thus, instances inside the genealogical tree are more fertile than other instances.

6.5 Beyond the Chilean Web

In this section we discuss how part of the results found for the Chilean Web can be generalized to the whole Web (or to other Web data sets). We are interested in estimating the number of children in a Chilean Web snapshot generated from parents outside Chile. We

have used a Web collection from Spain with 16.2 million pages, crawled in September 2004 [1]. We used the Spanish collection as the old collection and the Chilean 2005 collection as the new collection, and the same algorithms used for studying the Chilean Web.

We have found 11,800 new instances that are children from Spanish pages and from pages in the Chilean 2004 collection. These pages in the Spanish collection are either parents or children from the Chilean collection 2004. We have found 25,300 new instances in collection 2005 that are children only from Spanish pages. Thus, the total number of relations from Spanish pages is 37,100 instances. Collection 2005 has a total of 95,400 children from the Chilean collection 2004, considering intra-site plus inter-site relations. Comparing to the number of children from Spain, there are around two or three times more children from Chile than from Spain, in the Chilean 2005 collection.

In order to estimate the total number of children that may exist in the 2005 Chilean collection, we first estimate how big is the Spanish and Chilean Webs in comparison to all the Webs in Spanish spoken countries. We use the number of unique host names, which is measured by the Internet Systems Consortium[1]. We see that the Spanish spoken countries have a total 15.6 million host names, whereas Spain and Chile have 3.0 million and 745,000, respectively, representing 19.6% and 4.6% of host names in Spanish spoken countries.

A simple estimation is to consider that the other Webs from Spanish spoken countries (the other 75.6%, according to the number of host names) tends to generate the same number of children in the Chilean Web. In this case, there would exist 143,000 more children in the 2005 Chilean collection, considering the overlap with the 2004 Chilean collection, or 97,200 more children, excluding children from the 2004 Chilean collection.

This simple estimation does not take into account that there are other sites in Spanish language outside Spanish spoken countries, and that the Chilean Web also has pages in other languages. For these reasons we guess our estimation is a lower bound. Thus, the 2005 collection would have at least 217,900 children (95,400 from the 2004 Chilean collection, 25,300 from the Spanish collection, and 97,200 estimated for other Webs), which represent 23.7% of *new* instances in the 2005 collection. This percentage may also be valid for other Web data sets, as the Chilean Web has similar characteristics in comparison to other Webs [3].

7. GENEALOGY AND SEARCH ENGINES

In this section we start associating relations, especially parents, with metrics used by search engines to rank the results, and with the search engine results and click-through data. Our objective is to characterize the parents, which reflects the characterization of the user behavior when reusing content.

We carried out a series of experiments, presented in the following sections. In every case we compare data considering all the instances of a collection, considering only the intra-site parent instances, and considering only the inter-site parent instances (and sometimes the child instances too). Taking into account that intra-site relations are characterized by local reuse of the user's own Web site content, the metrics might present different results for intra-site and inter-site parents.

7.1 Genealogy and Pagerank

In this section we study the Pagerank [13] relevance measure for parents, children and instances in general. For clustered instances we chose the document of that cluster with the highest Pagerank, due to the fact that this document is probably the parent of the other duplicates in its cluster and it would be chosen by the search engine to be returned if its content match a query, eliminating duplicates in the answers. This heuristic for choosing the document to represent the cluster is also used for other experiments in the following sections.

Table 6 presents the average Pagerank for all the instances of the old collection, for parent and child intra-site and inter-site instances, for the adjacent collection pairs. The average for the collection pairs is also presented.

Table 6: Average Pagerank for old instances, parent instances and child instances. Values are multiplied by 10^5 for better visualization.

Collection pair	all	parents		children	
		intra	inter	intra	inter
02-03	0.082	0.070	0.080	0.022	0.034
03-04	0.029	0.027	0.052	0.022	0.048
04-05	0.032	0.027	0.081	0.021	0.020
05-06	0.033	0.038	0.042	0.021	0.019
average	0.044	0.040	0.064	0.021	0.030

The average Pagerank for child instances is very low, probably a consequence of the recent creation of the new instance. The average Pagerank for intra-site parents is very close to the average Pagerank for old instances (all instances). The average Pagerank for inter-site parents is quite high, on average 60% higher than for intra-site parents. This high difference indicates that parents are better connected on the Web graph than other documents, thus they are easier to be found than many other documents. In Section 7.3 we directly study the relationship between the search engine results and the parents.

Figure 6 presents the average Pagerank for the different components of the Web genealogical tree, that is, for original parents, old parents, children and parents, and sterile children. The first set of bars represents intra-site relations and the second set represents inter-site relations.

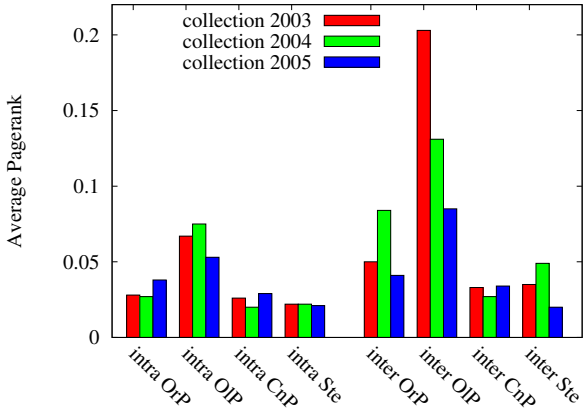

Figure 6: Average Pagerank for components of the Web genealogical tree. Values are also multiplied by 10^5.

The figure evidences that old parents have very high Pagerank. On the other hand, sterile children have low Pagerank.

[1]Internet systems consortium's domain survey, October 2007, http://www.isc.org/ds/

7.2 Genealogy and the Web Macro Structure

In this section we study how parents and children are connected in the Web graph macro structure [7]. We previously identified the Web macro structure component to which each document of a collection belongs, according to the Web macro structure component to which its site belongs. This heuristic is reasonable, given that by reaching a site, the user can also reach every document in that site.

Considering the average for all the five Chilean collections, tunnels, the island, the in, the out and the main macro structure components have 3.3%, 8.8%, 9.8%, 18.8% and 59.4% of the documents, respectively. Due to the large volume of data to be presented, we present together the out and main components, which are characterized by their connectivity, given that they are reachable from more pages.

Table 7 presents the percentage of connected components (main and out) for the whole old collection, and for intra-site and inter-site parents and children. Intra-site children have high connectivity because the child belongs to the same site of the parent, so that if the parent has high connectivity the child will have too. Intra-site parents have also high connectivity. This behavior may be due to the high volume of modifications in sites with more resources and more pages. As expected, the percentage of connected components for inter-site parents is higher than for inter-site children.

Table 7: Percentage of the Web macro structure connected components (main and out) for relations.

Collection pair	all	parents		children	
		intra	inter	intra	inter
02-03	74.9	90.3	85.2	89.8	83.3
03-04	72.9	88.2	90.5	95.3	93.3
04-05	82.6	93.5	87.1	91.1	73.3
05-06	81.9	89.0	91.3	87.4	67.1
average	78.1	90.3	88.5	90.9	79.2

Figure 7 presents the percentage of the Web macro structure connected components for elements of the Web genealogical tree. We see that inter-site child and parent instances are so weakly connected as sterile child instances (with an outlier for the 2004 children and parents).

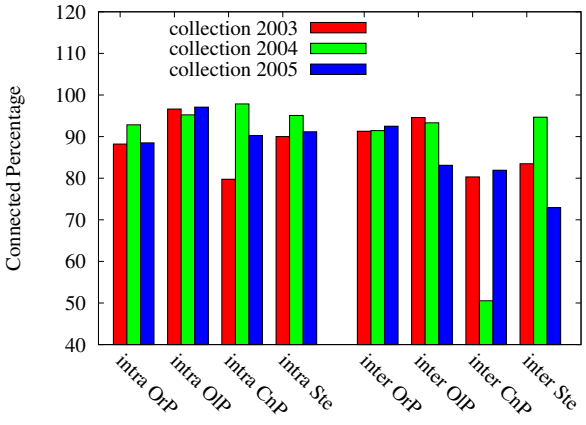

Figure 7: Percentage of the Web macro structure connected components (main and out) elements of the Web genealogical tree.

7.3 Genealogy and Query Results

In this part of the experiments we simulate a user performing queries in the past, and analyze click-through data. Initially, we observe whether the queries return the parents and how they are returned. The simulation is real because we used query logs and the same Web collection and query processor (we had access to the query processor as a black box) used by the search engine TodoCL. Given that this search engine was popular in Chile at that period, we try to associate queries (and clicks) to the parents, given that possibly part of the publishers of children used the TodoCL search engine (or other search engine whose ranking may not differ that much) in order to find content.

The query log we have available contains queries over a period of 10 months, from February to November 2004, and was applied to the collection 2004. The period of the logs starts one month after the collection 2004 was crawled and finishes two months before the collection 2005 was crawled (see Table 1). The one million most frequent queries were used and the top 5 results were considered. In this set, the most frequent of queries has 750,200 requests, whereas the least frequent query has 33 requests.

With the queries processed we used their results to compare how documents in general are returned, how intra-site parents and children are returned and how inter-site parents and children are returned. The children considered are for collection pair 2003-2004, that is, children in collection 2004. We perform the same study considering all the click-through data of the query log.

Figure 8 presents the average number of top documents returned per **occurrence** of queries (frequency of query is not considered), for documents in general (the first bar) and for components of the genealogical tree. On average a document is returned in 0.38 different queries. Given that the document is an inter-site original parent, on average the parent is returned in 0.87 different queries, which represents an increasing of more than 120% in relation to the average number of documents.

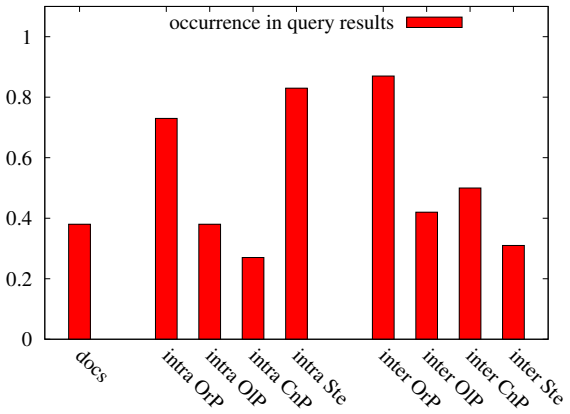

Figure 8: Average number of documents returned per occurrence of queries.

Figure 9 presents the average number of top documents returned for **all queries** (frequency is now considered), for documents in general and for components of the genealogical tree. For example, if a document d occurred in two queries A and B, submitted respectively 6 and 4 times, document d occurred for 10 requests in total. The intuition is that documents that appear more in results of queries are more likely to be copied.

We see that intra-site sterile documents are returned in a high

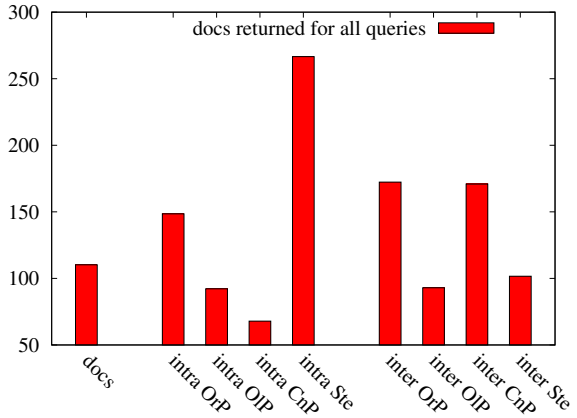

Figure 9: Average number of documents returned for all queries.

number of requests, which means that new documents with some old content from the same site may be relevant for a large set of requests. Comparing Figures 8 and 9, we see a similar behavior for each component. For instance, inter-site original parents and children and parents appears more frequently than old parents and sterile children.

Figure 10 presents the average number of **clicks** per document, for documents and for components of the genealogical tree. We see that intra-site original parents are frequently clicked, and that inter-site original and old parents are much more clicked than documents in general. Inter-site sterile documents have very low number of clicks.

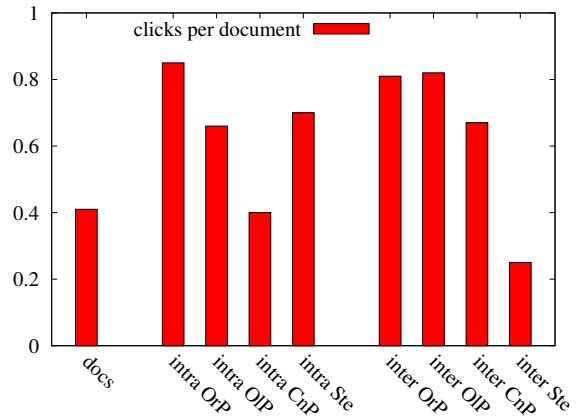

Figure 10: Average number of clicks per document.

For the three measures, in general we see that the values for inter-site parents are considerably higher than for documents in general, and also higher than for intra-site parents. These results represent evidence that part of the parents are associated to queries.

Characterizing the parents.

Considering this relation between inter-site parents and queries, we study the distribution of the frequency of parents in query results, with the goal of understanding whether the parents are the most returned set of documents or not. Our intuition is that the most returned documents are not the most copied documents. The most returned documents have normally a high Pagerank and not too much text. Maybe they are a good source of links for copied documents, and we guess that documents returned by queries and copied are returned by more specific queries rather than generic queries.

Figure 11 presents the distribution of documents according to the frequency with which they are requested in queries (the same measure used in Table 9), in a logarithmic scale. Figure 12 present an equivalent distribution, but only for inter-site parents.

Note that the axis range differs between Figure 11 and Figure 12. The frequency can be modeled as a power law, $\propto x^{-\theta}$. Note that every point plotted for parents is also represented as a point in the plot for parents, given that a parent is a subset of the document set, and the frequency of that document is obviously the same.

For instance, the last point at the bottom of Figure 11 means that one document was returned around 7 million times. The first point at the top means that a large number of documents were returned in only one query, which has the minimal frequency found in the log (33 requests).

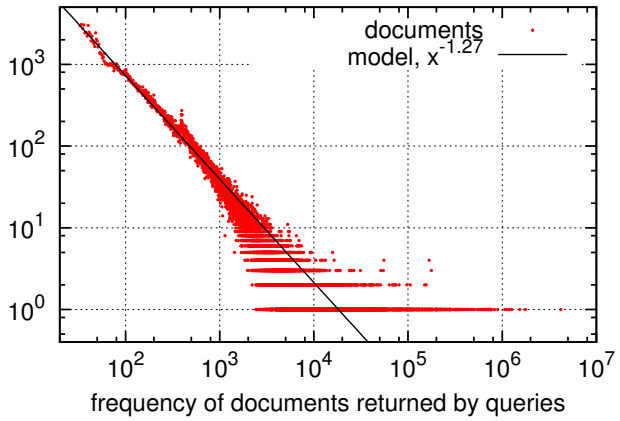

Figure 11: Distribution of documents in general returned by queries, according to their frequencies.

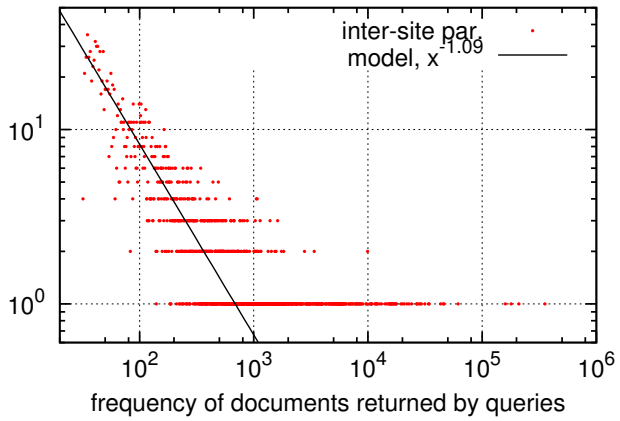

Figure 12: Distribution of inter-site parents returned by queries, according to their frequencies.

Comparing the general plot with the parents plot, we see that most of the parents in Figure 12 are represented with a low frequency in Figure 11. For example, the points in Figure 11 are concentrated between frequencies 1,000 and 10,000, while the points

in Figure 12 are more concentrated between frequencies 100 and 1,000. Figure 11 has many points after frequency 100,000. This is not the case in the distribution in Figure 12. Not only the range is smaller for parents, but also the power law has absolute exponent smaller than in the general case, showing that they are less spread.

Figure 13 presents together, the distribution of clicks on documents in general, and the distribution of clicks on inter-site parents. The same conclusions stated for the distribution of documents returned by queries are valid for the distribution of clicks. Observe that parents are not the most clicked documents.

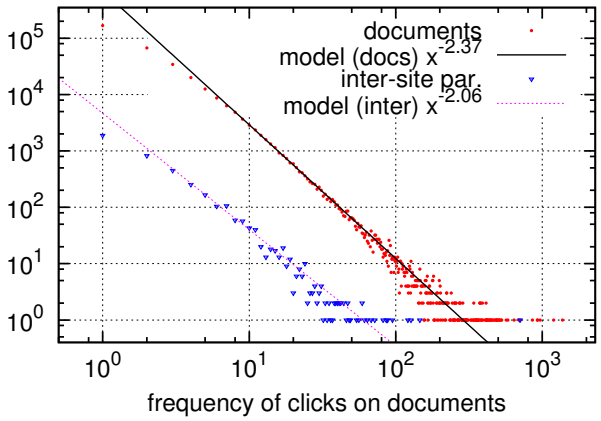

Figure 13: Distribution of clicks on documents in general and on inter-site parents.

The values presented in Figures 9 and 10 show that parents are returned in queries and they are clicked much more frequently than documents in general. At the same time, the plots show that the parents are not the most frequent documents returned by queries or the most clicked documents. These facts reinforce our intuition that part of the queries are used with the intention to copy a content, and in these cases the documents are not so frequent, probably because the query is more specific.

8. CONCLUDING REMARKS

In this paper we have investigated the evolution of textual content on the Web. We have shown that a significant portion of the Web content has been evolving from old content. We have presented estimations to generalize our finds to other Web data sets. We estimated that 23.7% of the new Web documents that appear within a span of a year have content from previously published documents, which is a high percentage. We also verified that previously copied pages are more likely to become parents again.

We have introduced the concept of genealogical tree on the Web, and studied its components. Basically, we have observed that inter-site parents have high pagerank, are well connected to the Web graph, appear frequently as result of real queries and are clicked frequently after a search. These results indicate that search engine ranking algorithms bias part of the Web content.

Acknowledgements

We would like to thank Graham Coleman for the English revision. 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), CNPq Grant 30.5237/02-0 (N. Ziviani) and CAPES grant 0694-06-1-PDEE (A. Pereira).

9. REFERENCES

[1] R. Baeza-Yates, C. Castillo, and E. N. Efthimiadis. Characterization of national web domains. *ACM Trans. Inter. Tech.*, 7(2):9, 2007.

[2] R. Baeza-Yates, C. Castillo, and F. Saint-Jean. Web dynamics, structure and page quality. In *Web Dynamics*, pages 93–109. Springer, 2004.

[3] R. Baeza-Yates and B. Poblete. Dynamics of the chilean web structure. *Computer Networks*, 50(10):1464–1473, 2006.

[4] T. Bennouas and F. Montgolfier. Random web crawls. In *16th Intl. Conf. on World Wide Web*, pages 451–460, Banff, Alberta, Canada, May 2007.

[5] K. Bharat and A. Broder. Mirror, mirror on the Web: a study of host pairs with replicated content. In *8th Intl. Conf. on World Wide Web*, pages 1579 – 1590, Toronto, Canada, May 1999.

[6] A. Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29, 1998.

[7] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Ninth International World Wide Web Conference (WWW'00)*, pages 309–320, Amsterdam, Netherlands, May 2000.

[8] J. Cho and S. Roy. Impact of search engine on page popularity. In *13th Intl. Conf. on World Wide Web*, pages 20–29, New York, USA, May 2004.

[9] J. Cho, N. Shivakumar, and H. Garcia-Molina. Finding replicated Web collections. In *ACM Intl. Conf. on Management of Data (SIGMOD)*, pages 355–366, May 2000.

[10] S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani. Topical interests and the mitigation of search engine bias. In *Proc. Natl. Acad. Sci. USA 103(34): 12684-12689*. National Academy of Science, 2006.

[11] A. Ntoulas, J. Cho, H. K. Cho, H. Cho, and Y.-J. Cho. A study on the evolution of the Web. In *US – Korea Conference on Science, Technology, and Entrepreneurship (UKC)*, pages 1–6, Irvine, USA, 2005.

[12] A. Ntoulas, J. Cho, and C. Olston. What's new on the Web? the evolution of the Web from a search engine perspective. In *13th Intl. Conf. on World Wide Web*, pages 1–12, New York, USA, May 2004.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the Web. Technical Report CA 93106, Stanford Digital Library Technologies Project, Stanford, Santa Barbara, January 1998.

[14] S. Pandey, S. Roy, C. Olston, J. Cho, and S. Chakrabarti. Shuffling a stacked deck: the case for partially randomized ranking of search engine results. In *Proceedings of the 31st Intl. Conf. on Very large Data Bases*, pages 781–792, 2005.

[15] TodoCL. www.todocl.cl or www.todocl.com, 2007.

[16] M. Toyoda and M. Kitsuregawa. What's really new on the web? identifying new pages from a series of unstable web snapshots. In *15th Intl. Conf. on World Wide Web*, pages 233–241, Edinburgh, Scotland, May 2006.

[17] J. Zhang and T. Suel. Efficient search in large textual collections with redundancy. In *16th Intl. Conf. on World Wide Web*, pages 411–420, Banff, Alberta, Canada, May 2007.