periodica polytechnica

Electrical Engineering
55/3-4 (2011) 119–126
doi: 10.3311/pp.ee.2011-3-4.04
web: http://www.pp.bme.hu/ee

RESEARCH ARTICLE

How much is involved in DB publishing?

I. Gyula Szabó

Received 2012-07-09

Abstract

XML has been intensive investigated lately, with the sentence, that "XML is (has been) the standard form for data publishing", especially in data base area.

That is, there are assumptions, that the newly published data take mostly the form of XML documents, particularly when databases are involved. This presumption seems to be the reason of the heavy investment applied for researching the topics of handling, querying and comprising XML documents.

We check these assumptions by investigating the documents accessible on the Internet, possible going under the surface, into the "deep Web". The investigation involves analyzing large scientific databases, but the commercial data stored in the "deep Web" will be handled also.

We used the technique of randomly generated IP addresses for investigating the "deep Web", i.e. the part of the Internet not indexed by the search engines. For the part of the Web that is accessed (indexed) by the large search engines we used the random walk technique to collect uniformly distributed samplings. We found, that XML has not(yet) been the standard of Web publishing, but it is strongly represented on the Web. We add a simple new evaluation method to the known uniformly sampling processes.

These investigations can be repeated in the future in order to get a dynamic picture of the growing rate of the number of the XML documents present on the Web.

Keywords

Database · XML · HTML · XHTML · Random walk sampling · Standardization

Acknowledgement

The research was supported by the project TÁMOP-4.2.1/B-09/1/KMR-2010-003 of Eötvös Loránd University.

I. Gyula Szabó

Department of Information Systems, Eötvös Loránd University, H-1117 Budapest, Pázmány Péter sétány 1/C, Hungary

1 Introduction

XML has been exhaustively investigated in the last ten years, with the reasoning "*XML ... has become the prime standard for data exchange on the Web*" [7], "*Organizations are increasingly using the world wide web to disseminate and distribute information. Most of this information is specified in XML which is emerging as the de-facto standard language for document representation and exchange over the Web.*" [10] (and much more . . .) That is, there are assumptions, that the newly published data take mostly the form of *XML* documents, especially when databases are involved. This opinion seems to be the reason of the heavy investment applied for researching the handling, querying and comprising *XML* documents [9]. In order to check these assumptions we estimate the weight of *XML* in the Web-publishing. When this weight will be determined, we can see, whether it complies with the "standard publishing method" statement. First of all, we should define the method of checking. All published data are to be accessed by the users of the Internet, i.e. each one of these data (each document) should be stored on a host and it should be accessible. That is, we have to measure the proportion of *XML* documents among the documents stored and accessible on the Internet. The simplest way to do this is determining the number of files with file-extensions specific for *XML* documents (e.g. .xml and .xhtml). We can query Google after these types of files using the parameter *filetypes:xml* and *filetypes:xhtml*. (A check for filetype:xml returned 1.409.000 documents in 2004 [6] by Google, it complies with Table 1 well).

As shown in Table 1 the absolute number of files with .xml extension is impressive, but the proportion to the number of .html-s is less: only 0.4%. The number of .xml files shows a rapid expansion until 2008, but since then, when Google shows the correct numbers, then the yearly increment has been stabilized about 2 million new*XML*documents; on the other side, the yearly increment of .xhtml-s is growing.

We have repeated this test three times in this year (2010): in June, July and August. The number of HTML documents has increased 100 million monthly (1.7%), while the decrement of .xml-s is about the same as the increment of .xhtml-s (about 8-900 thousand, 3.3% for .xml, 5.1% for .xhtml). Google gives

us a rough estimation over the number of documents found with the requested types, but only the first highest ranking 1000 files will be presented for accessing. By analyzing the returned documents, a lot of files with extension .xml are in fact HTML documents. (They are after all XHTML documents, stored on the HTTP-serving servers as Content-type:text/html). So, it is no use to build upon file extensions, when searching after *XML* documents, another search criterion should be chosen.

The growing rate for the number of .html files (i.e. Web pages) is obviously proportional to the growing rate for the number of hosts accessible on the Internet. Figure 1 shows the current state of the Domain Host Survey, made regularly by ISC ([16]). The growth is nearly linear: about 100 million new hosts yearly since 2001.

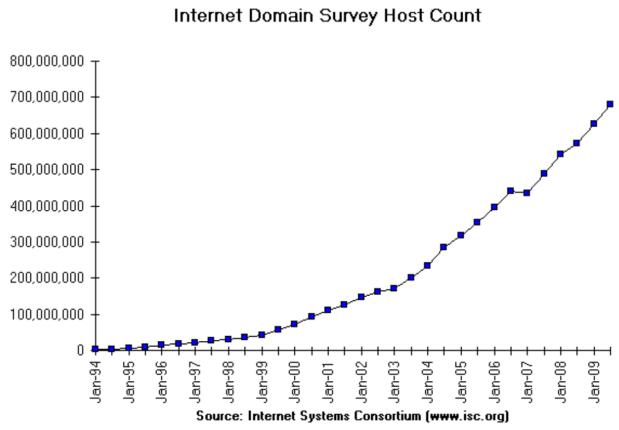

Fig. 1. Growth of domain host count

The goal of our discussion is to estimate the weight of *XML* documents on the Web. An interesting, connected topic is the popularity of the thema *XML* in the Web society. (We discuss this phenomenon in Section 7). An easy test to check the popularity of the topic *XML* is looking for the text "xml" by some search engines. Table 2 shows the results.

We can say that the topic *XML* is very popular in the Web society: a lot of talking is going on, but there is very little action.

Make another try before going on the harder way: ask again the four selected search engines after well typed *XML* documents, looking for the text "`<?xml version="1.0"`" (without the restricting *encoding* clause), that is, we would accept all *XML* documents, without concerning the language of publishing. Table 3 shows the number of results for the four largest search engines and for the two different *XML* versions announced until now. We have got, of course, a lot of descriptions over *XML*, training and lecturing materials also, but the total number of documents found (about 7.5 million by Google) is impressive, if you try to access each one of them, but among the billions of pages represented by the HTML files this number is not so significant.

The huge difference between the number of documents found by Google and Yahoo, compared to the results of MSN (40-50 million for Google and Yahoo, 7-8 million for MSN respectively) requires explanation. The Web pages accessed by a given search engine build up a domain of the Web, this is the indexable domain for this search engine. According to a research published in 2005 by Gulli and Signorini, [8], the estimated size of the indexable Web (the union of the indexable domains of the search engines) could be set to at least 11.5 billion pages as of the end of January 2005. They also estimated the relative size and overlap of the largest Web search engines. According to their estimation, Google was(is) the largest engine, followed by Yahoo, by Ask/Teoma, and by MSN. Google claimed to index more than 8 billion pages, MSN Beta claimed about 5 billion pages, Yahoo at least 4 billion and Ask/Teoma more than 2 billion. (the domains are overlapping). We used AltaVista as the fourth engine in the comparison, because Ask/Teoma hanged up. Google had to return the largest number of documents, when the count of answers would be proportional to the size of the indexable domains: it obviously does not appear so. From comparing the pairwise overlapping of the search engines's domains as given in [8], follows that the intersection between Google[Yahoo] and MSN is over 55%[49%] (proportion in Google([Yahoo]) or over 78%[67%] (proportion in MSN). That is, either Google and Yahoo index a large subdomain, full of *XML* documents and unaccessible by MSN at the same time, or, because there is no obvious reason to presume that MSN is biasing against *XML*, the result of searching by engines after *XML* documents is not reliable enough to measure the weight of *XML* documents on the wide Web. It happens, that either, the search engines use different search strategies, which deal very differently with *XML* documents, or, they return correct highest-ranked answers, but the total number of found documents is incorrect. In both cases, it follows, that we cannot use the search engines's results immediately for measuring the weight of *XML* documents on the Web. Anyway, the count of answers (documents found) decreases with time, according to each one of the four selected search engines.

Tab. 2. Searching after the text "xml" with search engines

Search engine	Count of answers
Google	416 000 000
Yahoo	1 220 022 035
MSN	75 300 000
AltaVista	1 220 000 000

We should try to test the Web otherwise.

2 Sampling the Web

There are a lot of methods for sampling the internet, we selected three thoroughly discussed procedures for trying:

- accessing uniformly distributed, random IP numbers
- random walking on the Webgraph
- focused sampling

Tab. 1. Searching the WWW for files with a given file type using *Google*

Searchcriterion	Count of answers	Growing rate since 1995
Filetype:xml	23 600 000 (Jun)	1995-2000: 480 000
	22 800 000 (Jul)	2001-2005:1 990 000
	26 500 000 (Aug)	2006: 1 820 000 2007: 2 220 000
		2008: 2 200 000 2009: 2 200 000
Filetype:xhtml	16 700 000 (Jun)	1995-2000: 314 000
	17 600 000 (Jul)	2001-2005: 999 000
	15 400 000 (Aug)	2006: 398 000 2007: 861 000
		2008: 1 880 000 2009: 2 210 000
Filetype:html	5 630 000 000 (Jun)	see Figure 1
	5 720 000 000 (Jul)	
	5 150 000 000 (Aug)	

Tab. 3. Searching the WWW for well typed XML documents using search engines

Version	Month	Google	Yahoo	MSN	AltaVista
1.0	June	43 770 000	51 800 478	8 430 000	19 300 000
1.0	July	43 600 000	51 800 595	8 410 000	19 100 000
1.1	June	1 022 000	23 400 223	7 410 000	26 800
1.1	July	1 010 000	23 300 228	7 690 000	27 100

Each one of these methods accesses Web pages to collect samples from them. Usually, a Web site consists of one or more Web pages, these Web pages are the visualisation of a set of documents (hosted on Web-servers), they can be displayed on the screen of a computer as an individual page. The main component of a Web page (adressed by a URL) was former generally a HTML file, nowadays the page can be based upon an XHTML document too. We define for our discussion the Web page as following:

Definition 1. *A Web page is a HTML or XHTML document, stored on a Web host and accessible by a URL on the World Wide Web.*

In the following discussion "the Web" refers to those subset of all Web pages defined above, which can be returned as a result of some HTTP GET request from a valid server on the Internet (including both static and dynamic Web pages; we consider those Web pages only which are hosted on servers supporting the HTTP protocol). We would say that a document *D* belongs to a page *w*, or the page *w* contains the document *D* iff when there is a hyperlink embedded in the main component of the page *w* referencing *D*. (We define the family of documents as the targets accessible by a URL on the World Wide Web).

In order to get an overall view of our sampling, let us give a formal definition. We denote by *W* the Web as defined above. We denote

Let $w \in W$, d a document, then

$$w \mapsto d \text{ iff when } w \text{ contains } dD \quad (1)$$

We settle that when

$$w \in W \text{ then } w \mapsto w \quad (2)$$

Let $F, G : W \mapsto \mathbb{N}_0$

be two weight functions defined on W. (3)

Let $S \subseteq W$ be the subset of Web pages selected by the sampling. Then let

$$q = \frac{\sum_{w \in S} F(w)}{\sum_{w \in S} G(w)} \quad (4)$$

be the proportion of the weight F relative to the weight G on S. When the sample is representative for the Web, q is a good estimation for the overall weight of F relative to G on the Web. For our goal let

$$F(w) = n, n \in \mathbb{N}_0 \quad (5)$$

for a given w iff when w contains exactly *n* XML documents, and let

$$G(w) = n, n \in \mathbb{N} \quad (6)$$

for a given w iff when w contains exactly *n* HTML documents ($G(w) \geq 1$ because of (2)).

Counting the number of the *XML* documents stored on the pages, we estimate the proportion of *XML*-documents among the Web pages in the whole Web. We need now a suitable definition of the functions F and G, that is, a passing and usable criterion of an *XML*(HTML) document.

An appropriate criterion can be based upon the http header attribute "Content-type". This data contains the pattern "xml" for each registered "Content-type" (see [15]) identifying an XML document which should be accessible by the conventional Web-browsers. So, we choose as selection-criterion for our XML-hunting that the string "xml" would be contained in the "Content-type" of the given document (e.g. Content-type=text/xml or application/xml or text/xhtml+xml etc.). For HTML documents we can apply the Content-type "text/html".

3 Sampling by random IP numbers

The simplest method for uniformly sampling the Web is selecting random, uniformly distributed IP numbers from the set of usable 32-bit numbers (restricting this selection for the allowed numbers as given in [12] and [13]). We can access the selected IP-s (the hosts addressed by them) using the HTTP protocol (command GET). A returncode 200 reports that the requested IP is currently active and accessible (it addresses generally a static Web server). This method assures a uniformly distributed sampling: the distribution of the generated IP-s is uniform, the selection is random (the generating algorithm guarantees this), and the assignment of IP-s to hosts is random regarding the numeric values of the IP numbers. Using this method, we can sample the whole Web: not only the highly indexed part, but the rest, the so called "deep Web" also. We used the GNU tool *wget* for sampling, slightly modified (a new parameter, -z *number* added, meaning generate *number* random IP-s, address them, and check if the IP valid, active, and the addressed host does contain *XML* documents). We tested with *number*=8000, occasionally checking, the proportion of the valid and active IP-s was at a stable 1% of the total accessed (generated and selected) IP-s. We made the sampling in two testphases: in the first testphase we collected the accessible IP numbers, i.e. the randomly selected and accessible hosts. The first phase of the test required 26 hours, and resulted 73 valid and active hosts visited (slightly less than 1% of the total 8000).

n the second phase we took the collected IPs as the starting set for accessing all pages hosted by them. These pages built up the subset S of W as defined above. We found $\sum_{w \in S} F(w) = 1$ and $\sum_{w \in S} G(w) = 820$, i.e. $q = 0.001$. The second phase required 15 hours execution time.

Chang et al. in [5] used this method to explore the deep Web, they used 1 million randomly generated IP numbers and it was a reasonable idea to follow their path for answering a simple query about the XML presence on the Web. But, only one visited page with XML on it is an unuseable result. It could be correct, though: Google has announced in July 2007, [14], that "recently, even our search engineers stopped in awe about just how big the web is these days – when our systems that process links on the web to find new content hit a milestone: 1 trillion (as in 1,000,000,000,000) unique URLs on the web at once!". Of course, the number of pages is less than the number of unique URL-s, but 1 XML from a 820 pages sampling is surely overrepresents the real weight of XML on the whole (indexed and deep) Web.

But, while we generated uniformly distributed random IP4 addresses, only 1% of them addressed valid and active Web hosts. 99% of the execution time has been wasted for trying to access unaccessible hosts. Moreover, there was only 1 page found containing XML documents. Using this method we cannot estimate the proportion of XML documents on the Web. This method was successfully used, as referred above, for another estimation [5]. The reason of the current unsuccessful test should yet be cleared: the referenced research could access 1 million IP-s under 240 hours, an accomplishment that was not reacheable for us.

4 Random walking on the Webgraph

Another ways for uniformly sampling the Web are based upon the graph structure of the Web, i.e. they are considering the Webgraph. Before discussing the uniformly sampling of pages using random walk on the web, we must first refine the definition of "the web" given in Section 2. There is a general accepted structure of the Web, as a directed graph, as suggested by Broder at al. in [4]. The nodes of this graph are the Web pages, and the edges are the hyperlinks referencing other (or the same) pages (they are normally contained in the HTML documents). The structure of the web is similar to that shown in Figure 2. According to this model the web graph divides into four parts of roughly equivalent size (see [2]):

1 A giant strongly connected component (the largest subset of nodes from which every pair of nodes are mutually reachable from one another by following links).

2 A "right" side containing pages reachable from (1), but which cannot reach (1) in return.

3 A "left" side whose pages can reach (1), but are not reachable from (1) in return.

4 All the rest (other small connected components, or pages that link to the right side or are linked from the left side).

We refer to the union of (1) and (2) as the "indexable" web, since this is the part that is easily explored by most users and search engines, and that arguably contains most of the meaningful content of the web. Our random walk and experiments are conducted mainly on this part of the web.

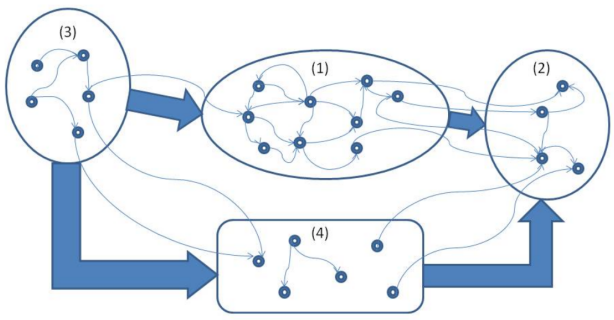

Fig. 2. The graph structure of the Web

A memoryless random walk on the Web graph is a Markov chain, where the states of this Markov chain are the nodes (i.e. Web pages) of the Webgraph and the transitions between states are realized by following the hyperlinks (edges) pointing to the

Tab. 4. Sampling the core using breadth-first search

Root	Duration	Robots	Pages visited	HTML	XML	Other
www.google.com	24h	off	28646	76958	1242 1,61%	46110
www.google.com	38h	on	20794	37154	916 2,46%	12863
www.msn.com	23h	off	37366	29011	378 1,01%	7977
www.msn.com	47h	on	44564	23465	342 0,76%	20757
www.yahoo.com	53h	off	39066	115901	1813 1,56%	9416

next node of the walk. So, each visit to a node results in one step of the random walk. We call a step a selfloop when the walk visits the same node in two consecutive steps of the walk (by traversing a loop to itself). According to a fundamental theorem of Markov chains, a random walk on an undirected, aperiodic and irreducible graph will converge to a unique stationary distribution. Once the walk reaches its unique stationary distribution, the probability of being in a node will not change although the walk takes more steps. On such a graph a random walk converges to a unique stationary distribution where the probability of being in a node is proportional to its degree. That is, the random walk converges to a unique stationary and uniform distribution, when the graph is undirected, aperiodic, irreducible and regular.

Sampling the Web by random walking on the Web graph has been throughly discussed in the literature, the proposed models start from a node in the core and follow the edges (links) as described above. Bar-Yossef at al. ([1]), Henzinger at al. ([3]) used a model of undirected, irreducible and regular graph. The natural Web is, of course, directed and irregular, because the in-links to a page are not known, and the degrees of the nodes are very different. The used models tried to improve the case, claiming in-links from search engines and using selfloops to make the graph regular. The random walk will be so a Markov chain on a connected, undirected, regular graph. Walks with these properties can be proven to eventually approach a uniform distribution over the edges of the graph. The referenced sampling methods proposed a two-phases model:

1 collecting a large sample by random walking (using improvements: adding selfloops or in-links)

2 subsampling the saved collection to reach uniform distribution (smoothing higher degrees or page ranks)

The referenced methods could produce an almost perfect, unbiased sampling of the Web, though the sampling could not realize totally the assumptions of the convergence to the uniform stationary distribution. But, for our goal, to estimate the weight of XML documents on the Web, a simpler, one-phase processing seems to be adequate: random walk on the core, starting from a high-degree node.

The random walk on the natural Webgraph (directed and irregular) can converge to a uniform distribution, but it can fail also: the walking can stuck on a node. We have not implemented a perfect model complying with the referenced propositions, first of all while they all used either a generated graph or a large sample from the Web collected by others and not the living Internet. Moreover, there is no known number of random steps to reach (or to approximate) the stationary distribution.

As a warming up, we have begun sampling the Webgraph using a simple BFS (breadth-first search): the outcome of this process would be a perfect sampling, the collection of all Web pages (of the core, when the root of the processing stands in it) in the end, but, of course, in an acceptable time we can reach a relative small domain only. We have executed a couple of BFS samplings, the results are summarized in Table 4.

We computed the proportion of the number of XML documents relative to the number of the HTML documents as defined in (5) and (6). The results show no definite correlation in the proportion of XML-related documents when tested with robots (i.e. omitting the documents that are not suggested for accessing by search engines), so we used both settings (robots on/off) in the following tests too to check if such a correlation does exist. A strong correlation between the weight of the XML documents, measured with robots on/off would suggest a different handling of XML/non XML documents by Web administrators. (e.g. preferring XML documents for administrative, insider handling etc.)

We have improved the sampling:

- added a burn-in phase using only host-out links (no BFS!)
- tested with different burn-in periods
- used a random walk after burn-in phase
- repeated all tests with switched robots on and off

We have found that a burn-in period of 10.000 steps has improved the uniformity of the random sampling, while a too short period or without burn-in phase at all the random walking can stuck in a close, neighbouring environment of the root.

Table 5 shows the settings of the random walks:

- set checking against robots.txt according to column "Robots"
- start from the home site given in column "Root"
- follow host-out links for a preset number of steps (column "B-I", as burn-in time)
- in the "burn-in" phase, connect leafs ("dead-end" pages without hyperlinks) random to a page already visited

Tab. 5. Sampling the core using random walking on the Webgraph

Root	Duration	Robots	B-I steps	R-W steps	Pages	Files
www.google.com	21 h	off	10000	130100	9836	18048
www.msn.com	54 h	off	1000	93361	n.a.	30747
www.msn.com	47 h	on	0	63146	n.a.	44564
www.yahoo.com	20 h	off	10000	237057	4305	12930
www.altavista.com	18 h	off	10000	193855	4346	19189
www.altavista.com	45 h	on	10000	212199	6652	16330
www2.lib.udel.edu	14 h	on	10000	24528	n.a.	7134

Tab. 6. Results of the random core sampling

Root	Robots	Files	HTML	XML	Other
www.google.com	off	18048	10678	212 1,98%	7158
www.msn.com	off	30747	29935	350 1,17%	462
www.msn.com	on	44564	23465	342 1,45%	20757
www.yahoo.com	off	12930	9635	599 6,21%	2696
www.altavista.com	off	19189	13951	866 6,20%	4372
www.altavista.com	on	16330	13577	1013 7,46%	1740
www2.lib.udel.edu	on	7134	6824	89 1,30%	221

- process random walking (the column "R-W" shows the number of executed steps)
- run the test until a total time of "Duration" hours

The column "B-I steps" shows the number of steps used in the burn-in phase (in this phase, we selected a link - i.e. an edge - from a parents out-links that points to a child-node on another host, when possible), the column "R-W steps" reflects the number of random steps (we allowed self-loops in this phase, i.e. hyperlinks pointing to the same page would be followed). The column "Pages" displays the number of different pages visited during the last 10.000 steps of the random walking. The column "Files" gives the total number of documents collected from those pages. We didn't use in-links and didn't added empty selfloops in order to make the graph regular as proposed by Bar-Yossef at al. [1] and Henzinger at al. [3]. But we checked occasionally the set of visited pages and found, that their set has stabilized after approximately 20-30 hours of execution, so we aborted the execution at this point and evaluated the testresults.

For evaluation we need to interpret the set of visited pages and their distribution. Let S denote the set of Web pages in the core, and let $|S| = N$ (i.e. $S = \{w_1, \ldots, w_n\}$) and let $P_{i,j} (1 \leq i, j \leq N)$ the transition propability matrix of the Webgraph. Because the core is irreducible and aperiodic (we settled in (2) that each page has a self-loop) then the random walk converges to a stationary distribution

$$\pi = \{\pi(i), \ldots, \pi(n)\} \text{ associated with P such that}$$

$$\pi(j) = \sum_{j=1}^{N} \pi(j) P_{j,i} \text{ and } \pi(i) = 1/M_i,$$

where M_i is the expected return time.

Let $s_0 \in S$ (a page in the core) the root of the random walk and S_0 the set of visited pages $S_0 \subset S$ (while S is irreducible) and let K the number of random steps, then $|S_0| \leq K$ because the pages could be visited more than once. The frequency of a page $w_0 \in S_0$ let K_i where the index i identifies the page $w_0 = w_i \in S$. Then $K_i/K \to \pi(i)$ when $K \to \infty$.

Rusmevichientong et al. [11] proposed an algorithm Directed-Sample for uniformly sampling the Webgraph as a directed graph by compensating the unequal frequencies of the pages by the selection for visiting them. We have chosen another solution: we executed a random sampling until a (more or less) stationary state and compensated the different page-frequencies in the evaluation. That is, let K the number of random steps (i.e. the number of visited pages) in the walk, $S_0 = \{w_1, \ldots, w_K\}$the visited pages, and let N_K the number of different pages, so $N_K \leq K$. Let $S'_0 = \{w'_1, \ldots, w'_{N_K}\}$ the list of the different pages in our sample of visited pages, $K_i (1 \leq i \leq N_K)$ the frequency of w'_i in S_0. We can compensate the unequality of frequencies when we take in account each different page in the sample only once by the evaluation of their attributes, while $N_K \to N = |S|$ when $K \to \infty$. Let $F(w)$ a function defined on the Web pages as given in (3). Then

$$\lim_{K \to \infty} \frac{\sum_{w \in S'_0} F(w)}{N_K} = \frac{\sum_{w \in S} F(w)}{N}$$

$$\text{Let } q_K = \frac{\sum_{w \in S'_0} F(w)}{\sum_{w \in S'_0} G(w)} \text{ and let } q = \frac{\sum_{w \in S} F(w)}{\sum_{w \in S} G(w)}$$

then it follows that $\lim_{K \to \infty} q_K = q$.

We used from the random walk only the pages visited during the last 10.000 steps to make up the sample for the evaluation, because our test executions reached about 20.000 random steps and we wanted consider the second half of the walking (in order to improve the uniformity of the distribution). We collected all documents contained in the pages of the sample, as we executed a second, preparing phase for the evaluation, by visiting

the N_K selected pages again and collecting the contained documents. Let again S'_0 the list of the different pages in our sample of visited pages, let $D = \{d|w \mapsto d, w \in S'_0\}$ (Column "Pages" in Table 5 reflects $|S'_0|$, while column "Files" in Table 5 and in Table 6 show $|D|$).

Using the set D we computed $\sum_{w \in S'_0} F(w)$ according to (5), (first part of column "XML" in Table 6) and $\sum_{w \in S} G(w)$ (column "HTML" in Table 6, and computed q according to (4)), (second part of column "XML" in Table 6).

Table 6 shows these computed results of the random walks starting from the home sites of the largest search engines and from a home page collecting links to bioinformatical data bases. The column "Files" shows the number of the collected documents after the burn-in phase, the columns "HTML","XML","Other" reflect the number of the documents of "Content-type: text/html", "Content-type" complying with the ones given in [15] as *XML*-related types, and the rest, respectively.

As cited previously,[8], the large search engines are strongly interconnected and they access huge domains of the core: subdomains of google and yahoo have been strongly represented in the random walking path started from each roots. The high q values for the rows with "Root" yahoo and altavista need explanation: all three of them have caught RSS feeds and walked in them for a long time, this part of the random walk is responsible for about the half of XML documents found. But it means, that *XML* is represented really strongly in specific subdomains of the Web, i.e. it is worth trying a focused sampling.

5 Focused sampling

A focused sample is a uniformly chosen sampling from a thematically unified subdomain of the Web. In fact, focused sampling has been used generally for collecting data related to topics of social aspects. But, when we redefine the goal of our discussion, as "estimating the amount of production made by the community of people regularly publishing XML documents on the Web" then the domain of their target could be analyzed using the methods of focused sampling.

We assume, that this domain is strongly interrelated, so a random walk started from a page inside it would stay in the domain for a long time, and collect a lot of XML-containing pages. We queried Google again, looking for the text "`<? xml version="1.0"`", took the first 1000 answers (there were 265 different pages among them) and set up a random walk with the following algorithm:

1 start with the first page of the list of Google answers

2 follow the out-links using random jumps recursively

3 when stuck or dead-end found take the next page from the list of 1.

4 go back to 2.

The algorithm can be essential improved by implementing a function for retrieving hyperlinks from an XML document (we parsed only the .html files for hyperlinks when building up the list of outgoing edges from the currently visited node). But we could collect a large set of documents, and their distribution shows a strong representation of XML documents, i.e. XML publishing seems to behave as a community-building topic.

Tab. 7. Results of the focused sampling

Duration	21 h
Pages	50457
Files	26604
HTML	19139
XML	928 4,85%
Other	6537

Table 7 shows the result of the focused sampling, we found essential more individual XML documents with this process (we check a lot of them) than found by the random walking on the core (the larger proportions would be caused by rss feeds). But we cannot estimate neither the size of the XML-focused domain nor the real weight of XML documents in it currently, because of the lack of appropriate methods for following the hyperlinks embedded in XML documents.

6 XML and Data Bases

Our preliminary goal was to investigate the weight of XML as a tool for Web data base managament. We have begun our investigation by selecting a couple of Web sites of bioinformatics dealing with large data sets and many users. We found, that they prefer the using of XML documents, but they don't do it always explicitly, the pure statistical measurement of the files on the sites shows a very low proportion of XML documents (s. Table 8). Browsing the sites manually, we can state that XML as phenomenon is present, the older databases have been converted to XML from the original relational DB, the new data entries are mostly required to be in XML form, a schema is proposed.

7 Conclusion

XML has not(yet) been the standard form of the publishing on the World Wide Web: neither the total, estimated proportion of XML documents worldwide (about 2 % relative to the number of HTML documents) nor the growth rate makes it to the standard tool. Continuing the simple measurement presented in Table 3 to the level of a longer time serie, can help developer, researcher selecting an appropriate subject of their efforts. But the presence of XML is well established: there are a lot of services based upon XML (e.g. rss feeds) and the thema XML is in heavy discussion among developers of tools, drafts, descriptions. We can say that the thema XML is very popular in the Web society: a lot of the documents found during our investigation were training materials, lecturing texts, examples and drafts dealing with the possible applications of XML, less using XML

Tab. 8. XML documents on some sites of bioinformatics

Home page	# XML docs	% XML docs	# all files
gmod.org	40	4.11	9717
helix.nih.gov	1	0.01	2208
molgen.biol.rug.nl	0	0	866
www.ebi.ac.uk	72	0.21	33092
www.genomesonline.org	1	0.62	161

for a real task. XML will be nowadays intensive searched, lectured, explained. It seems, that the online Data Bases build upon XML lately, especially when collecting new data by user of their services. There exists an XML-oriented community, in order to estimate the amount of their production one should be able to walk on the focused domain of XML documents, i.e. select and follow hyperlinks embedded in XML documents. There is no appropriate method yet known for us, we would like to implement one and repeat the focused sampling as described in Section 5. Random walking on the Webgraph is a suitable method for uniformly sampling the Web, our evaluation method (take the last visited n pages once in account) is a simple, but effective method to answer aggregate queries concerning the whole Web.

References

1 **Bar-Yossef Z, Berg A C, Chien S, Fakcharoenphol J, Weitz D**, *Approximating aggregate queries about web pages via random walks*, International Conference on Very Large Databases (VLDB) (2000), 535–544.

2 **Bar-Yossef Z, Kanungo T, Krauthgamer R**, *Focused sampling: Computing topical web statistics (Approximating aggregate queries about web pages via random walks)*, Technical report, IBM T.J Watson Research Center (2005).

3 **Baykan E, Henzinger M, Keller S F, De Castelberg S, Kinzler M**, *A Comparison of Techniques for Sampling Web Pages*, 26th International Symposium on Theoretical Aspects of Computer Science STACS 2009 (2009), 13–30.

4 **Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S, Stata R, Tomkins A, Wiener J**, *Graph structure in the web: Experiments and models*, Proceedings of the Ninth Conference on World Wide Web (May, 2000), 309–320.

5 **Chang K C C, He B, Li C, Zhang Z**, *Structured databases on the web: Observations and implications*, Report UIUCDCS-R-2003-2321 (Feb, 2003).

6 **DuCharme B.**, *Googling for XML* (2004), `http://www.xml.com/pub/a/2004/02/11/googlexml.html`.

7 **Fan W., Libkin L.**, *On XML Integrity Constraints in the Presence of DTDs*, Journal of the ACM (JACM) **49** (May 2002), no. 3, 368–406.

8 **Gulli A, Signorini A**, *The indexable web is more than 11.5 billion pages*, Proceedings of WWW 2005 (2005), 902–903.

9 **Leighton G, Barbosa D**, *Optimizing XML Compression*, Proceedings of the Sixth International XML Database Symposium (XSym 2009) (2009), 91–105.

10 **Ray I, Muller M**, *Using Schemas to Simplify Access Control for XML Documents*, Lecture Notes in Computer Science (2004), no. ISSU 3347, 363–368.

11 **Rusmevichientong P, Pennock D M, Lawrence S, Giles C L**, *Methods for sampling pages uniformly from the world wide web*, AAAI Fall Symposium on Using Uncertainty Within Computation (2001), 121–128.

12 `http://en.wikipedia.org/wiki/IP_address#IPv4_private_addresses`.

13 `http://en.wikipedia.org/wiki/Multicast_address`.

14 `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`.

15 `http://www.iana.org/assignments/media-types/`.

16 `http://www.isc.org/solutions/survey`.