**RJ10339 (A0502-003) February 2, 2005**
**Computer Science**

# IBM Research Report

## Focused Sampling: Computing Topical Web Statistics

**Ziv Bar-Yossef, Tapas Kanungo, Robert Krauthgamer**
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099

**Research Division**
**Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich**

# Focused Sampling: Computing Topical Web Statistics

Ziv Bar-Yossef Tapas Kanungo Robert Krauthgamer

IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120
Email: {ziv,kanungo,robi}@almaden.ibm.com

November 17, 2003
(Reformatted January 31, 2005)

**Abstract**

Aggregate statistical data about the web is very useful in numerous scenarios, such as market research, intelligence, and social studies. In many of these applications, one is interested not in generic data about the whole web but rather in highly focused information pertinent to a specific domain or topic. Furthermore, timely acquisition of this information provides a competitive advantage.

Focused statistical data can be gathered by a brute force crawl of the whole web, or by a "focused crawl", which collects mainly pages that are relevant to the topic of interest. Crawling, however, is an expensive enterprise, requiring substantial resources. For the purpose of gathering statistical data, random sampling of web pages is a much faster, cheaper, and even more reliable approach. We develop the first efficient method for generating a random sample of web pages relevant to a given user-specified topic. Previously, techniques for getting only an unfocused sample of pages from the whole web were known. Our method is based on a new random walk on (a modified version of) a subgraph of the web graph.

# 1 Introduction

Consider an international company, which would like to break into emerging markets in Asia. A representative sample of web pages from the target countries could be an excellent starting point for understanding their business culture and market needs. Focused statistical data about domains and communities on the web could be of great importance also to business development, intelligence, and social and political studies.

A *focused sample* is a uniformly chosen sample of web pages from a thematically unified community (TUC) of pages. We will typically refer to broad themes, which consist of million of pages. These themes can be diverse, ranging from domain names, through pages written in a specific language, to pages relevant to a given topic.

**Motivation.** Apart from the applications mentioned above, focused sampling could be an important addition to the web data miner toolbox. Calculating degree distributions or finding the distribution of subtopics within a topic can be done easily with a random sample, without the need to fetch millions of pages and perform calculations over massive data sets.

Evaluating the recall of a focused crawl is a well-established open problem [15]. Focused sampling resolves this problem by allowing one to estimate the fraction of the focus pages that are covered by the focused crawl. In more generality, a focused sample can be used to form an objective mechanism for evaluating the topic-specific coverage of general purpose search engines. This may give rise to a richer set of techniques for comparing the quality of search engines (rather than the blunt total coverage numbers search engines use to boast at nowadays).

**Our contribution.** Selecting a uniformly chosen sample of web pages about a given topic can be carried out either by a full-fledged crawl or by a focused crawl, which guides towards on-topic pages. However, crawling, and even focused crawling, are formidable tasks, requiring significant investments in infrastructure, bandwidth, and software engineering. Moreover, crawlers and focused crawlers typically prioritize fetching pages with high quality and PageRank, and thus may not be suitable for generating a uniform, unbiased, sample of pages.

We develop the first efficient method for generating a focused sample of web pages relevant to a specified topic. Previously, techniques for only getting an unfocused sample of pages from the whole web were known [6, 25, 5, 39]. In principle, an unfocused sample from the entire web induces also a random sample of topic-relevant pages. However, such a scheme is not practical for the purpose of focused sampling. For example, in order to obtain $n$ samples from a community that constitutes 0.2% of the web (which is still a large number!), we would need to produce $500n$ unfocused samples. This makes the sampling procedure prohibitively inefficient. It should be noted that no search engine currently provides a service of generating a random page from a subset of their index, or even from their entire index.

Our focused sampling algorithm is much faster and cheaper than focused crawling; it requires a significantly smaller number of web pages to be fetched, and can be implemented on a desktop PC. Focused sampling is even a more reliable approach for gathering statistics, because it is performed over a shorter period of time. This makes focused sampling less vulnerable to changes in web pages during its execution, effectively reflecting a "snapshot" of the web.

**Methodology.** The sampling method we develop is based on a new random walk on the graph formed by web pages and their hyperlinks. We employ a variant of the random walk of Bar-Yossef *et al.* [5], where the walk is performed on the *undirected* graph, a strategy that relies on a search engine, such as Altavista [1], to provide the backlinks of a page. But in order to obtain a sample of the focus data, we restrict the walk to regions of the web that contain relevant pages, an approach that is inspired by the focused crawling literature.

The simplest way to do this is to restrict the walk to the subgraph induced on the pages relevant to the focus topic. However, such a hard focus rule might provide inaccurate results if the subgraph is not well-connected (i.e., has low conductance). This may happen when the subgraph consists of two or more subcommunities that are in small interaction with each, such as the pro-life and pro-choice groups within the abortion topic. In such cases, employ a soft focus rule that "extends" the subgraph on which the random walk is performed so that it contains also

some pages at the frontier of the topic. This is supposed to increase the subgraph's connectivity, but at the same time it dilutes the fraction of relevant pages in the subgraph, which decreases the fraction of relevant pages traversed by the walk. To keep the sampling practically efficient, we should limit the extent to which the walk may depart from the relevant pages.

We perform extensive experiments with both the hard focus procedure and the soft focus procedure. We demonstrate that for "well-developed" topics the two approaches are doing equally well and come close to generating a uniform sample of pages. For less connected topics, the soft focus rule is doing better due to its flexibility. We also show that our algorithms are robust to classification errors. Using a high quality classifier to determine whether a given page belongs to the topic or not has a marginal effect on the overall quality of the sample.

The rest of the paper is organized as follows. We start with an overview of related work in Section 2. In Section 3 we formally define the focused sampling problem, provide the necessary background from the theory of random walks, and describe our hard focus and soft focus algorithms. In Section 4 we describe implementation details about how our system works in practice. Section 5 includes an extensive overview of the experiments we performed as well as discussion of the lessons learned from them. Section 6 ends with concluding remarks.

## 2 Related work

Our work draws upon research done in areas of focused crawling, sampling, and text classification. In this section we first describe the various approaches adopted by researchers in these domains and then distinguish our approach from the past work.

**Focused crawling.** Focused crawling was introduced by Cho, Garcia-Molina, and Page [16] and Chakrabarti, van Der Berg and Dom [14, 15]. Cho *et al.* used properties like in-degree and anchor text keywords to guide a crawl towards relevant pages. Chakrabarti *et al.* proposed a semi-supervised learning algorithm to idenitfy on-topic pages. They also introduced the notions of "hard focus rule" and "soft focus rule", referring to two possible startegies to guide the crawl to further on-topic pages. Diligenti *et al.* [18] suggested a sophisticated focused crawling algorithm, which uses the "context" of a page to determine whether it is a good gateway for discovering more pages about the topic. This context consists both of the link-induced neighborhood of the page and of its content-based model. Rennie and McCallum [37] used a reinforcement learning approach to crawling the web.

All the above algorithms are aimed at fetching as many quality pages as possible relevant to the focus topic. They are not designed to generate a random sample from these pages.

**Sampling and random walks.** Bharat and Broder [6] used random queries to estimate the coverage and the overlap between search engines. Henzinger *et al.* [24] invented a random walk algorithm, which converges to Google's PageRank [2, 34, 7] distribution over the nodes of the web. They then modified the random walk [25] so it approximates a nearly-uniform distribution over the web. Bar-Yossef *et al.* [5] proposed a random walk on an undirected and regular version of the web graph as means of generating near-uniform samples of web pages. Rusmevichientong *et al.* [39] extended the above approaches to handle both directed and undirected graphs.

The above schemes are all aimed at generating an unfocused sample of pages. They cannot be used to efficiently collect a focused sample.

Kanungo *et al.* [30] used a topical sample of web pages to discover what fraction of images on the web contain textual information. However, the sample was generated by querying Google, and thus the returned pages did not have a uniform distribution (since mainly pages with high PageRank are returned). Moreover, this sample relies on the freshness of the Google repository, which may not provide an updated snapshot of the web.

Davison [17] and Menczer [32] studied the rates of divergence from a topic if one starts an unguided random walk from an on-topic node. Their results indicate that performing a random walk that stays focused is a non-trivial task.

**Web data mining.** Kumar *et al.* [31] described an algorithm for mining implicitly-defined web communities. The key idea was to search for small bipartite cores as signatures for web communities. Broder *et al.* [10] study various properties of the web graph and in particular show it has a "bowtie" structure. Dill *et al.* [19] demonstrate that the same global structural properties of the web graph appear also in its subgraphs, which are specified by themes, topics, or geographical proximity. Chakrabarti *et al.* [12] study the "topic distribution" of the web.

**Classification.** Supervised and unsupervised statistical classification have been studied in the literature for several decades. Jain *et al.* [26, 27] provide a good survey of the topic. Text books by Duda and Hart [20], Fayyad *et al.* [21], Fukunaga [22], and Ripley [38] cover the topics of supervised and unsupervised classifiers, neural nets, and statistical decision trees. The C4.5 decision tree classifier, which we used in our algorithm, is described in detail by Quinlan [36].

**General background.** A book by Charabarti [11] dicusses crawling, sampling, and web data mining. Baeza-Yates and Ribeiro-Neto [4] cover topics on text information retrieval. Broder and Henzinger [8] provide an excellent survey of of information retrieval algorithms on the web.

# 3 The focused sampling algorithm

In the discussion below, "the web" refers to the collection of all HTML pages that can be returned as a result of some HTTP GET request from a valid server on the Internet (including both static and dynamic HTML pages). We denote by $W$ this collection of pages. The "web graph" is a directed graph $G = (W, E)$, whose vertex set is $W$ and whose edges corresponds to the hyperlinks on pages in $W$. For a page $w \in W$, we will denote by $N_{\mathrm{OUT}}(w)$ the "out-neighbors" of $w$—all the pages that are pointed to by hyperlinks in $w$. Similarly, $N_{\mathrm{IN}}(w)$ denotes the "in-neighbors" of $w$—all the pages that contain hyperlinks pointing to $w$.

To simplify our arguments, we freeze an instantaneous snapshot of the web, and describe the algorithms as operating on this static entity. Since our algorithms run in a relatively short amount of time (measured in hours), we ignore issues arising from the fact that the web is dynamic.

## 3.1 Problem definition

Let $P : W \to \{0, 1\}$ be a Boolean predicate. Let $S \subseteq W$ be the subset of web pages selected by the predicate. That is, $S$ consists of all the web pages $w$, for which $P(w) = 1$. Intuitively, $P$ is a query or a theme, and $S$ is the set of pages pertinent to this query/theme.

Our goal is to design an algorithm that generates uniform samples from the set $S$. Specifically, given an input parameter $n$, the algorithm is required to output $n$ uniformly and independently chosen pages from $S$. In practice, it would suffice to generate a uniform random sample of $n$ pages from a large subset $S' \subseteq S$.

The algorithm assumes that $W$ and $P$ satisfy the following properties:

1. Given a URL of a page $w \in W$, the algorithm can fetch the text of $w$.
2. Given a page $w \in W$, the algorithm can get the URLs of its out-neighbors and its in-neighbors.
3. Given a page $w \in W$, the algorithm can determine whether $P(w) = 1$ or $P(w) = 0$.

We should note right away that even this type of limited access to $W$ and $P$ is not always realistic. Fetching a page given its URL and getting the URLs of its out-neighbors are standard. However, there is no natural way of obtaining the URLs of all the in-neighbors of a given page. Determining the exact value of $P(w)$ may be also tricky, depending on the definition of $P$. If $P$ selects all the pages about a specific topic, for example, then determining $P(w)$ requires classification of $w$, which may incur errors. In this section we shall consider the ideal model, in which $N_{\mathrm{IN}}(w)$ and $P(w)$ can be extracted from $w$. In Section 4, we describe how our algorithms are adapted to address the more realistic scenario.

### 3.2 Random walks on graphs

Our algorithm performs a random walk on the web graph in order to generate the random samples from $S$. We now give a brief overview of the key facts about random walks needed to describe and analyze our algorithm.

Let $G$ be an undirected graph on $N$ nodes. A random walk on $G$ is a stochastic process that continuously visits the nodes of $G$ in some random order. The walk starts at the some node $u \in G$, and at each step chooses one of the neighbors of the currently visited node uniformly at random to be the next node to visit.

A probability distribution over the nodes of $G$ is specified by a non-negative vector $\mathbf{q}$ of dimension $N$, whose entries sum to 1. A random walk on $G$ is formally described by an $N \times N$ probability transition matrix $\mathbf{M}$. $\mathbf{M}$ is a stochastic matrix (i.e., its rows are probability distribution vectors). The $(u, v)$ entry of $\mathbf{M}$ contains the value $1/d(u)$, where $d(u)$ is the degree of the node $u$. Given an initial probability distribution $\mathbf{q}_0$ (which is typically concentrated on a single node of the graph, which we call the "starting node"), the random walk induces probability distributions $\{\mathbf{q}_t\}_t$ for each step $t = 0, 1, 2, \ldots$. Then $\mathbf{q}_{t+1}$ is the product $\mathbf{q}_t \cdot \mathbf{M}$.

A standard fact from the theory of random walks and Markov chains (see, for example, [33]) is that if $G$ is connected and non-bipartite, then the sequence $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \ldots$ has a unique limit distribution $\pi$, which is independent of the initial distribution $\mathbf{q}_0$. Moreover, the form of $\pi$ is well-understood: it assigns to each node $u$ a probability that is proportional to its degree (specifically, $\pi(u) = d(u)/2|E|$, where $|E|$ is the number of edges in $G$). In particular, when $G$ is a regular graph (i.e., all the nodes have the same degree), the limit distribution $\pi$ is uniform over the nodes of $G$.

Thus, random walks provide a convenient way to generate random samples from the limit distribution, using only local information about the graph. The key element missing in order to guarantee an efficient sampling procedure is the analysis of the convergence rate to the limit distribution (which is termed in the Markov chain literature the "mixing time").

One particularly useful characterization of the mixing time of random walks on undirected graphs is in terms of the "spectral gap" of the matrix $\mathbf{M}$. The spectral gap is the difference $\sigma = |\lambda_1| - |\lambda_2|$, where $\lambda_1, \lambda_2$ are the two eigenvalues of $\mathbf{M}$ with the largest absolute value. A classical folklore result from the theory of Markov chains shows that the number of steps $t$ required until $\mathbf{q}_t$ is very close to $\pi$ is at most $O(\frac{1}{\sigma} \cdot \log N)$ (the constant is very small). That is, the larger the spectral gap, the faster the random walk converges to its limit distribution.

Random walks on $G$ give us thus the following scheme for generating $n$ random samples from the limit distribution $\pi$: run $n$ independent random walks on $G$ starting from arbitrary nodes (the starting nodes could be the same) for $\tau = O(\frac{1}{\sigma} \cdot \log N)$ steps each. Take the final nodes reached by these random walks as the sample points.

Aldous [3] proposed a somewhat more efficient sampling scheme, if the goal of the samples is to estimate the selectivity of a predicate $P : G \to \{0, 1\}$ (that is, the ratio $|S|/|G|$, where $S \subseteq G$ is the set of nodes selected by $P$). He uses a *single* slightly longer random walk of length $\tau' + n$ steps, and takes the last $n$ nodes visited by the walk as the sample points. Gilman [23] and Kahale [29] show that a delay of $\tau' = O(\frac{1}{\sigma} \cdot \log N \cdot \frac{|G|}{|S|} \cdot \frac{1}{\epsilon^2} \log(1/\delta))$ suffices, if one would like to get an estimate of the selectivity $|S|/|G|$ to within an additive error of $\epsilon$ and with confidence $1 - \delta$.

### 3.3 Unfocused sampling

Our algorithm builds on the WebWalker algorithm of Bar-Yossef *et al.* [5], which generates a near-uniform unfocused sample of pages from the web. Let us quickly overview their algorithm.

Bar-Yossef *et al.* [5] create a random walk on a connected, undirected, non-bipartite, and regular version of the web graph. The graph is made connected by focusing on the largest strongly connected component (SCC) of the web graph and on the part that is reachable from it (OUT). (The "bowtie" paper of Broder *et al.* [9] shows that $W' = SCC \cup OUT$ constitutes about half of the web.) The graph is made undirected by ignoring the directions of hyperlinks,

and it is made non-bipartite and regular by adding weighted self loops to each node of the graph—that is, if $D$ is some large number that is guaranteed to be higher than the degree of any node on the web, then each node $u \in W$ is added a self loop of weight $D - d(u)$.

By the discussion in the previous section, this random walk is guaranteed to converge to a uniform limit distribution on the nodes of $SCC \cup OUT$. [5] also used a large crawl to estimate that the spectral gap of the probability transition matrix of their random walk is $\sim 1/10^5$. Thus, assuming that the size of the web $N$ is at most $10^{10}$, the mixing time of the random walk is at most 4 million steps. However, most of the steps of this walk are spent in the artificial self loops and since these steps are "free" (do not require any communication over the network), the actual number of steps needed in practice is about 100.

### 3.4 Hard focus sampling

We next describe our first algorithm for generating a random sample of pages from the set $S$ selected by the predicate $P$. This algorithm is a straightforward adaptation of the WebWalker algorithm to the new context.

Recall that $W$ denotes the web, $G$ denotes the web graph, $P$ is a predicate, and $S \subseteq W$ is the set of pages selected by $P$. Let $G_S$ denote the subgraph of $G$ spanned by the nodes in $S$. The vertex set of $G_S$ is $S$ and the edges of $G_S$ are the edges of $G$ that connect two nodes in $S$. For a node $u \in S$, let $N^S_{\mathrm{IN}}(u)$ and $N^S_{\mathrm{OUT}}(u)$ denote, respectively, the sets of in-neighbors and out-neighbors of $u$ in $G_S$. Note that $N^S_{\mathrm{IN}}(u) = N_{\mathrm{IN}}(u) \cap S$ and $N^S_{\mathrm{OUT}}(u) = N_{\mathrm{OUT}}(u) \cap S$. Let $d_S(u)$ denote the degree of $u$ in $G_S$; that is, $d_S(u) = |N^S_{\mathrm{IN}}(u)| + |N^S_{\mathrm{OUT}}(u)|$. Let $D$ be any large integer that is bigger than $d_S(u)$ for all $u \in S$.

Our hard focus sampling algorithm runs the WebWalker algorithm on the graph $G_S$. The random walk starts at some node $s \in S$, which we know belongs to the SCC of $G_S$. (In practice, we can choose some central node on the topic specified by $P$; e.g. the top hit in Google.) When visiting a node $u \in S$, with probability $\alpha = 1 - d_S(u)/D$, the random walk stays in $u$ (i.e., uses the artificial self loop). With probability $1 - \alpha$, the random walk chooses a page $v$ in $N^S_{\mathrm{IN}}(u) \cup N^S_{\mathrm{OUT}}(u)$ uniformly at random and visits $v$. (Note that the set $N^S_{\mathrm{IN}}(u) \cup N^S_{\mathrm{OUT}}(u)$ is always non-empty, because the random walk got to $u$ through an edge from some neighbor that belongs to $S$.)

If the $SCC \cup OUT$ part of $G_S$ is large, then the discussion in Section 3.2 implies that this random walk converges to a uniform distribution over a large subset of the nodes of $G_S$. Recent experiments of Dill *et al.* [19] indicate that the web is "self similar". That is, the same global structural properties of the web as a whole emerge when considering various subgraphs of the web induced by different themes or topics. In particular, Dill *et al.* showed that for "well-developed" topics, such as "golf" or "mathematics", the corresponding subgraphs exhibit the desired "bowtie" property (i.e., that $SCC \cup OUT$ is a large fraction of $G_S$). We conclude then that for such topics our hard focus random walk algorithm will generate a random sample of pages from a large fraction of the set $S$.

An issue that still needs to be addressed is what would be the convergence rate of the hard focus random walk. We do not have estimates of the spectral gap of $G_S$ for different choices of $S$. However, we note that the spectral gap of the random walk's transition matrix is a *structural property* of the graph $G_S$. It is closely related to the "conductance" of the graph [28], which captures how many "bottlenecks" or "isolated parts" the graph has. Extrapolating from the results of Dill *et al.*, we conjecture that: i) the spectral gaps of subgraphs corresponding to well-developed topics are similar to the spectral gap of the whole web graph, and ii) the hard focus random walk will produce the near-uniform samples rather quickly on these subgraphs.

### 3.5 Soft focus sampling

We next turn to the situation in which the graph $G_S$ does not have a bowtie structure or good conductance—the two key properties required to guarantee the hard focus random walk convergence quickly to a uniform distribution over a large subset of $S$.

Two extreme methods for generating samples from $S$ are the following. The first is to collect uniform samples from the whole web $W$ via an unfocused random walk, and then use only the samples that belong to $S$. Since the samples from $W$ are uniform, also those of the samples that belong to $S$ are uniform on $S$. The main drawback, however, is that we may need to run the unfocused random walk for a very long time until we collect a sufficient number of samples from $S$. In fact, we expect only one in $|W|/|S|$ of the unfocused samples to belong to $S$. If $S$ is a small fraction of the web (say, 1%) then the length of the random walk could be prohibitive.

The other extreme is the hard focus random walk. Here we are guaranteed that all the samples belong to $S$, but as mentioned above they may not be distributed uniformly in $S$ or in a large subset of $S$.

These two extremes exhibit a tradeoff between what we call the "sample precision" and the "sample recall". The sample precision is the fraction of the samples generated that belong to the set $S$. The sample recall measures how uniform or "representative" the samples that belong to $S$ are. The hard focus random walk has a good precision but a poor recall, if the topic $S$ is not "well-developed". The unfocused random walk has a high recall but a low precision.

Our key idea for dealing with subgraphs $G_S$ that are not "well developed", is to try to take the good from both the unfocused random walk and the hard focus random walk. Instead of running the random walk on the graph $G_S$ (hard focus) or on the graph $G$ (unfocused), we run the random walk on an intermediate graph $G_T$, where $S \subseteq T \subseteq W$. $T$ needs to be chosen carefully so as to ensure both sample precision and sample recall. $T$ should not be too large, so that the ratio $|T|/|S|$ is small enough to achieve a good precision. $T$ should not be too small, so that the graph $G_T$ will possess the bowtie and conductance properties required for the random walk to converge quickly to a uniform distribution. We call this approach "soft focus sampling".

There may be many possible ways to create such an intermediate set $T$. In this paper we consider two methods: the "topical neighborhood" method and the "trial-and-error" method.

**The topical neighborhood method** In the topical neighborhood method the set $T$ consists of all the pages in $W$ that either belong to $S$ or that are reachable from $S$ through an undirected path of at most $k$ links. In other words, $T$ is the (undirected) neighborhood of $S$ of radius $k$. Note that the size of the neighborhood grows very quickly with $k$, so in order to keep the sample precision high we need to choose $k$ to be very small. In our experiments, we choose $k = 1$. In this case $T$ is simply all the nodes in $S$ and their neighbors.

The soft focus random walk using a topical neighborhood of radius works as follows. The walk starts at some node $s \in S$. After visiting a node $u \in T$, with probability $\alpha = 1 - d_T(u)/D$, the random walk stays at $u$. With probability $1 - \alpha$, the random walk chooses uniformly at random a neighbor $v \in N^T_{\text{IN}}(u) \cup N^T_{\text{OUT}}(u)$, and visits $v$.

Two technical complications that come up when trying to perform this random walk are: testing whether a node $u$ belongs to $T$ or not, and figuring out the degree $d_T(u)$ of a node $u \in T$ may require too many fetches. A brute force algorithm for performing these tasks would need to fetch all the nodes in the neighborhood of $u$ of radius $k$ and check how many of them belong to $S$. We solve this mini-problem by using random walks again. We run $B$ short random walks of length $k$ to estimate what fraction of the pages at distance $k$ from $u$ belong to $S$. If at least one of these shorts walks lands in a page in $S$, then $u$ belongs to $T$. The number of the walks that end at pages in $S$ provides a crude approximation of $d_T(u)$. In our experiments we set $B = 10$ and $k = 1$, and thus this procedure amounts to fetching only 10 neighbors of $u$.

If $G_T$ indeed possesses the bowtie and conductance properties, then the random walk on the topical neighborhood $T$ is guaranteed to converge to a uniform distribution over a large subset $T' \subseteq T$. If the intersection $S \cap T'$ constitutes a large fraction of $S$, then the random walk has a good recall on $S$ as well. Finally, if $S \cap T'$ is a large fraction of $T'$, then the random walk has a high sample recall.

**The "trial-and-error" method** The "trial-and-error" method is much faster to run than the topical neighborhood method. However, this method lacks the theoretical guarantees of precision and recall. In this method we allow the random walk to wander away from the set $S$ for some number of steps $k$. If after $k$ steps, we have not seen a page in $S$, we backtrack to

the last page visited that belongs to $S$ and try again.

More formally, the random walk keeps a counter $C$, which counts the number of steps since the last time it visited a page in $S$, and a pointer $p$ to that page. The random walk starts at some page $s \in S$. After visiting a page $u$, the random walk chooses uniformly at random a neighbor $v \in N_{\text{IN}}(u) \cup N_{\text{OUT}}(u)$. Note that $v$ is chosen not only from the neighbors of $u$ in the subgraph $G_S$, but from the full list of the neighbors in the graph $G$. If $v \in S$, the random walk visits $v$, resets the counter $C$ to 0, and sets $p = v$. If $v$ is not in $S$ and if $C < k$, the walk visits $v$ and increments $C$. If $C = k$, the walk "backtracks" to the node pointed by $p$ and continues as before.

For the "trial-and-error" method, we cannot rigorously define the set $T$ on which the random walk is made. Intuitively, this random work tries to approximate the behavior of the random walk on the topical neighborhood while using far fewer fetches.

A third possible method for getting an intermediate graph $G_T$, which we did not pursue in this paper, is via hierarchical classification. Suppose we have a hierarchical taxonomy of topics, such as the one represented by the Yahoo directory. If $S$ is the set of pages relevant to some node $c$ in this hierarchy, we can define $T$ to be the set of pages relevant to some ancestor $c'$ of $c$.

# 4 Concrete implementation

In the description of our random walk algorithms we made some unrealistic assumptions, such as the ability to get all the in-neighbors of a page or the ability to determine the value of the predicate $P$ exactly. In this section we describe the concrete implementation of our algorithms, which tries to approximate the behavior of the "ideal" algorithms.

**Obtaining in-neighbors** In order to get the list of in-neighbors of a given page $w$ we use two sources: the first is the random walk itself. We store on disk all the pages visited by the random walk and all their incident links. If any of the stored pages has a link to $w$, we know they are in-neighbors of $w$. At first glance, this may seem a negligible source of in-neighbors, but it turns out to be an extremely valuable one. Since the pages the random walk visits are correlated, the walk is likely to visit in-neighbors of $w$ before it visits $w$. Moreover, the only way to visit $w$ is to first visit a neighbor of $w$; in many cases this is an in-neighbor of $w$.

The second source of in-neighbors is search engines. Most search engines allow one to query for all the pages that contain a hyperlink to a given URL. In our experiments we use the search engine AltaVista [1]. The results from the search engine do not provide a complete solution to the in-neighbor problem. First, search engines themselves cover only part of the web and therefore may miss some of the hyperlinks. Second, for pages with very large in-degree, like Yahoo, search engines allow access to just a small fraction of the results. In the experiments, we took only the first 100 results returned by the search engine in order to minimize the number of fetches.

Like in [5], our random walk algorithms maintain consistency by making sure that if a node is visited several times, the list of neighbors available to the walk is the same in all the visits. That is, the first time the walk visits a node $w$, it stores the set of neighbors it is aware of on disk (these are the out-neighbors and all the in-neighbors from previously visited nodes and from the search engine). If the walk happens to visit $w$ again it chooses only from the list of neighbors that is stored on disk. Note that it is possible, even likely, that after the walk visits $w$ for the first time, it encounters other in-neighbors of $w$ it has not been aware of before. The walk ignores the hyperlinks from these new in-neighbors when it visits $w$ again.

The above consistency requirement guarantees that the random walk is performed on a well defined graph. This graph is a (random) subgraph of $G_S$ (in the case of hard focus sampling) or of $G_T$ (in the case of soft focus sampling). The consistency also reduces the natural bias the random walk has toward nodes that have high in-degree or PageRank. As noted in [5], such nodes are likely to be discovered early in the walk, but will not be revisited again more frequently than others after they have been visited once.

**Computing the predicate** Depending on the semantics of the predicate $P$, it may or may not be easy to compute. In our experiments we consider two types of predicates: URL predicates and topical predicates. A URL predicate is some condition on the structure of the URL of the page. For example, we considered all the pages whose url belongs to the `.uk` domain. A topical predicate is given by a keyword or a collection of keywords that specify a theme or a "topic". In the experiments we considered the topics "cycling", "abortion" and "HIV/AIDS".

Testing whether a page satisfies a URL predicate is usually trivial, requiring no more than matching a regular expression. Checking whether a page is relevant to a topic or not is much more intricate, requiring classification. In principle, the classification error may interfere with the random walk algorithm and cause is to diverge to pages not in $S$. In the experiments we use two classifiers: a naive classifier, which just checks whether the topic keywords appear in the page or not, and a standard decision tree based classifier (C4.5 [36]). We compare the results obtained from running the focused sampling algorithms with the two classifiers in order to evaluate how sensitive the algorithms are to the accuracy of the classifier.

The C4.5 classifier is trained using a corpus of web pages with on-topic and off-topic class labels. The on-topic set is created from the set of documents returned by a Google search. The off-topic set is created from the documents that are not traversed in a preliminary focused crawl using the naive classifier. The documents are then tokenized and converted into lower-case. Each document is represented as a vector where each vector component is a term's Term Frequency Inverse Document Frequency (TFIDF) [4] score. If a term does not exist in a document, its TFIDF score is zero. The length of the vector is equal to the total number of unique terms in the training corpus. The vectors are further normalized to length 1 prior to providing it to the classifier. The form of our TFIDF scoring function is adopted from that used by Chakrabarti *et al.* [13].

**Avoiding traps and bottlenecks** In order to avoid crawler traps or just large sites that contain few links to the rest of the world (such as the amazon sites), our algorithms make sure the random walk does not spend too much time consecutively within the same host. This is achieved as follows: the algorithm maintains a sliding window of length $L$, consisting of the hosts of the last $L$ pages visited by the random walk (together with the frequency of each host). If the new node chosen to be visited makes the frequency of one of these hosts exceed $L/2$, then the algorithm ignores the new node and jumps to a random node in the history of the walk. In our experiments we set the window size to be $L = 100$. This approach makes sure that even if the random walk is bouncing back and forth between two hosts, it will detect the infinite loop, and jump outside.

The choice of the random page to which the algorithm jumps is done as follows. The algorithm maintains the last $H$ distinct hosts visited by the random walk and the last pages visited on each of these hosts. When there is a need to jump to a random node, one of these host is chosen at random and the algorithm visits the last page visited on that host. The reason to choose from the list of distinct hosts rather than simply from the history of the walk is that the latter approach causes reinforcement of "trap hosts", in which the walk already spent many steps.

# 5 Results and Discussion

## 5.1 Experimental setup

We ran numerous experiments aiming at four goals: (1) evaluating the quality of our focused sampling approach; (2) comparing accuracy and performance the different focused sampling algorithms; (3) estimating the effect of the choice of the classifier on the quality of the focused sample; and (4) use focused sampling to discover new and interesting facts about the web.

In order to achieve these goals we ran the three focused sampling algorithms (hard focus, topical neighborhood, and trial-and-error) on four different themes: pages whose url belongs to the `.uk` domain, pages about cycling, pages about abortion, and pages about HIV/AIDS.

These topics represent a variety of different communities on the web. Abortion is a polarized community, while cycling is more cohesive. `.uk` is a large and versatile domain, while the rest are more focused.

For each of the four topics, for each of the three focused sampling algorithms, and for each of the two classifiers, we performed three runs of the random walk. The length of each run was either 10,000 actual steps or 12 hours, whichever came first. (Note that the number of fetches in each run could be much larger than 10,000, especially in the two soft focus algorithms.) In the soft focused sampling algorithms, the number of actual samples generated by each run was much less than 10,000, since many of the visited pages were off-topic. We thus selected from the visited pages those that are on-topic, discarded the first 1000, and took the remaining pages as our samples.

We ran the experiments on two machines, each having a pair of AMD 1.6 GHz processors, 3GB of main memory, 180GB hard disk, and 100Mbit/Sec network interface that is connected to the Internet through a firewall.

Our principal benchmark for evaluating the quality of the focused samples is a large crawl (consisting of more than a billion pages) that was made available to us in IBM Almaden. Since we had a full index of these crawled pages, we could easily generate a uniform sample of pages from the crawl, and compute various statistical properties of pages relevant to the above mentioned themes. Since the crawl is rather comprehensive, we used the statistical values derived this way as the "ground truth". We then compared the values obtained from the samples generated by our focused sampling algorithms against the "ground truth" values, in order to get indications for the quality of the samples.

The statistical properties we have chosen to compute are the distributions of the following: the out-degrees of on-topic pages, the domain name of on-topic pages, the fraction of intra-topic hyperlinks of on-topic pages (i.e., hyperlinks that point to other on-topic pages). An additional statistic is the fraction of on-topic pages that are relevant to some sub-topic.

## 5.2 Evaluation experiments

In the first set of experiments we evaluate the quality of the samples generated by the three focused sampling algorithms by contrasting statistics derived from them with statistics derived from the large crawl. In all of these experiments we consider only the runs that use the simple keyword-based classifier.

Figure 1 shows the distribution of out-degrees of pages in the `.uk` domain and of pages about cycling, as predicted by each of the four sets of samples. Figure 2 depicts the distribution of domain names among pages in the `.uk` domain and among pages about cycling. Figure 3 shows the fraction of cycling pages that are relevant to the subtopic "mountain biking".

Notice that in Figure 2 the distribution of the domains in each crawl looks very similar. However, to quantify the similarity, we conducted a statistical $\chi^2$ test [35] to test the null hypothesis that the two binned distributions are same. The test could not reject the null hypothesis at a 0.05 significance level for each of the three pairs of distributions, formed by comparing our crawl against each of hard, trial & error, and topical neighborhood.

There are a number of conclusions that one can draw from the results of these experiments. The first is that for both topics and for all the focused sampling algorithms, the statistics obtained from the focused samples provide a reasonable approximation of the real statistics. For example, the largest deviation in the estimation of the largest domain was less than 16% in the cycling experiment and less than 14% in the `.uk` experiment. We expect these deviations to be even smaller if we would have run the random walks for a longer time.

The second conclusion is that when the topic is well-developed and densely connected (like the `.uk` domain) the behavior of the focused sampling algorithms is more similar to each other, and closer to the ground truth. This is very apparent from the out-degree distribution, which is narrowly concentrated in the u̇k experiment and more scattered in the cycling experiment.

The third conclusion is that the trial-and-error algorithm surprisingly seems to give the closest results to the truth. It beats both the hard focus algorithm and the topical neighborhood

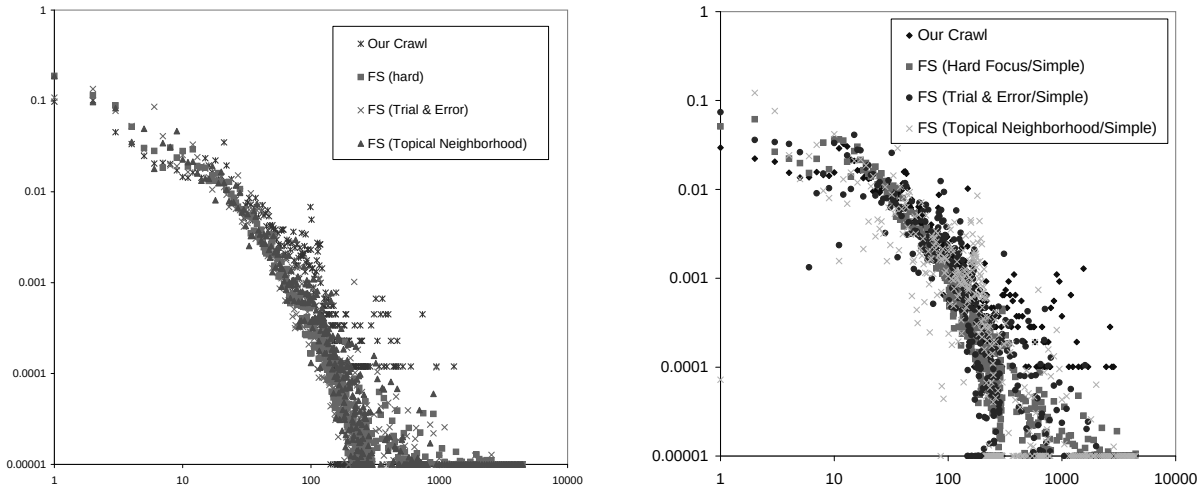


Figure 1: Out degree distributions for the `.uk` (left) and cycling (right) topics, as computed from a crawl and by focused sampling.

algorithm in both experiments. One possible explanation for this is the following. The hard focus algorithm is too conservative and thus may miss opportunities to accelerate its convergence to the uniform distribution by visiting off-topic pages en route to new on-topic pages. The topical neighborhood algorithm is too inefficient and thus, in many cases had a poor yield of samples. The scarcity of samples immediately affects the quality of the statistical estimations. It seems that the trial-and-error algorithm is doing the best, because it is more flexible than the hard focus random walk and yet efficient enough to yield a large number of samples.

### 5.3 Comparison experiments

The next set of experiments is used to compare different configurations of the algorithms. First, we wanted to measure the effect of the classification error on the quality of the focused sample. We compare two runs of the hard focus sampler; one uses the simple keyword-based classifier (which is very crude, thus incurring a high classification error) and the other one uses the decision tree classifier C4.5 (which is assumed to have a lower classification error).

The first experiment shown in Figure 4 measures the fraction of pages within the abortion topic that contain the terms "pro-choice", "pro-life", only one of the two, and both of them. The second experiment describes the fraction of out links in cycling pages that lead to other cycling pages. The height of the $i$-th column specifies the fraction of cycling pages, for which $10i$ percent of their out-links point to other cycling pages.

Both experiments indicate that the effect of the classification error is mild. In particular, a high classification error rate does not cause the focused sampling algorithm to lose the focus and diverge to other topics.

Next, we measured the sample precision of each of the focused sampling algorithms. That is, what fraction of the pages they visit actually belong to the focus topic. The precision of the hard focus algorithm is by definition 100%. The average precision of the topical neighborhood algorithm turned out to be 70%, while the average precision of the trial-and-error algorithm was 26%. We conclude that the precisions of all our algorithms are reasonable, requiring a number of random walk steps which is no more than four times the desired number of samples.

### 5.4 Data mining experiments

Our last set of experiments aimed at exploiting our new mechanism to explore new facts about the web. Figure 5 compares the domain name distribution and the fraction of intra-topic

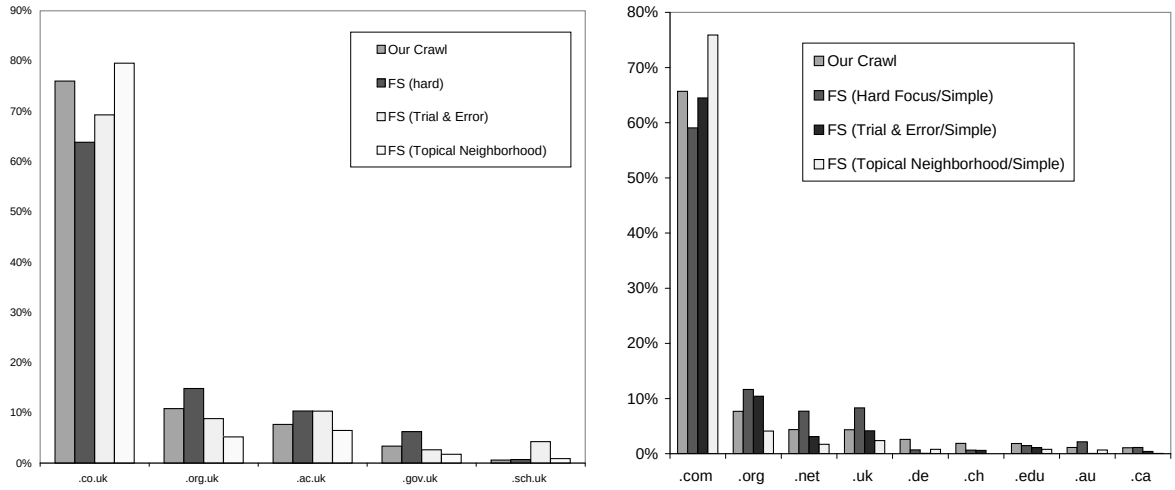


Figure 2: Domain name distribution for the `.uk` (left) and cycling (right) topics, as computed from a crawl and by focused sampling.

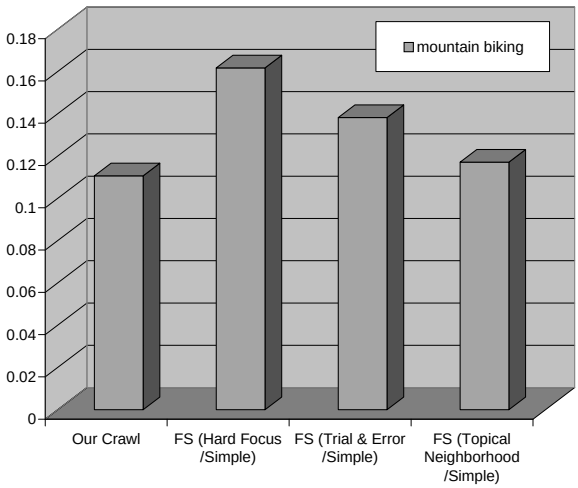


Figure 3: Frequency of the mountain biking subtopic within cycling, comparing the crawl to three focused sampling methods.

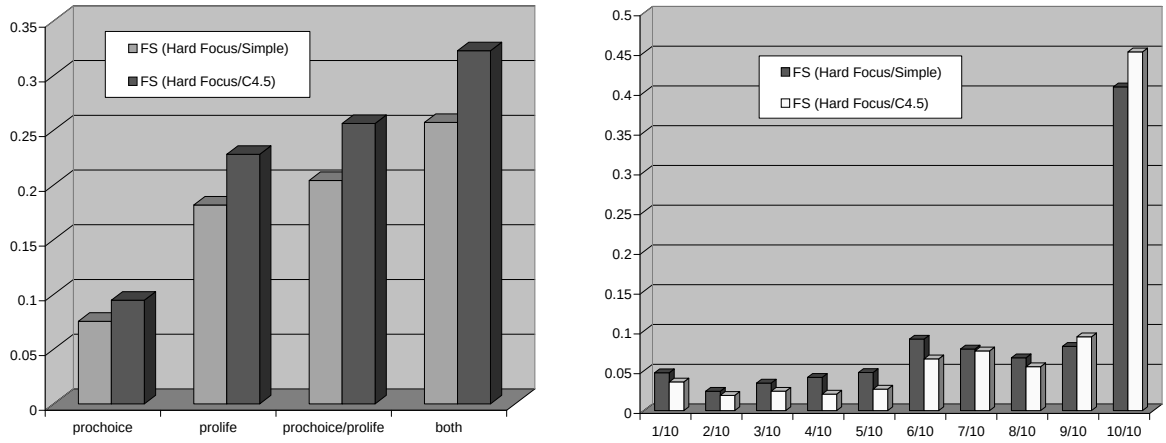


Figure 4: Comparison between the simple keyword-based classifier and the C4.5 classifier, when applied with the hard focus rule. The left picture plots the frequency of some subtopics among abortion pages. The right picture plots the distribution of the fraction of on-topic outlinks among cycling pages.

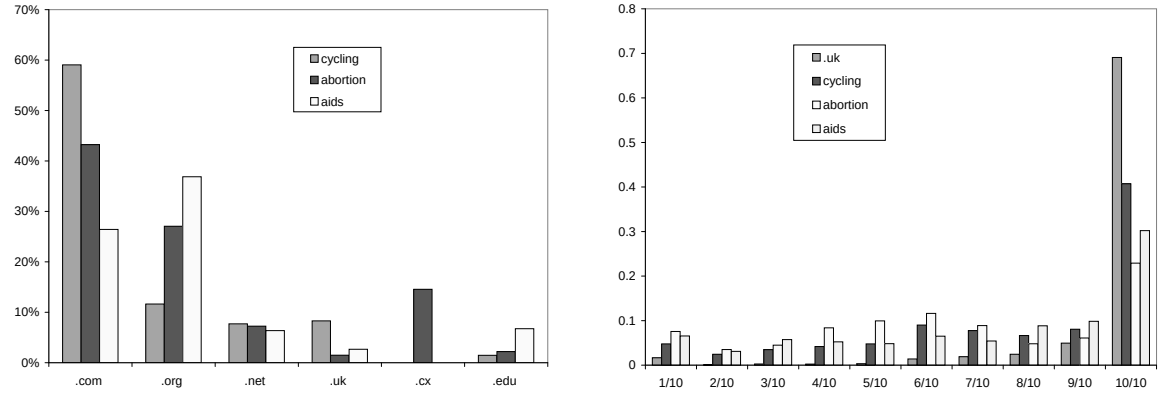


Figure 5: Characteristics of different topics: the distribution of domains (left) and that of the fraction of on-topic outlinks (right), both computed using focused sampling with a hard focus rule and a simple keyword-based classifier.

hyperlinks among the four topics considered.

The intra-topic out link distribution clearly indicates that `.uk` is a more segregated theme than the rest of the three. This is not surprising since `.uk` consists largely of British sites and is confined to a well-defined geographical region, while the others are more universal topics that relate well to other domains and topics.

Another interesting phenomenon is reflected by differences in the domain names among the other topics, cycling, abortion, and HIV/AIDS. Cycling seems to be a highly commercialized hobby, since almost 60% of its pages belong to the `.com` domain. HIV/AIDS is a more humanitarian theme, occupying non-profit organizations.

Finally, we picked two small samples, each consisting of 102 pages: one from the `.uk` samples of the hard focus random walk and the other from the `.uk` samples of the trial-and-error random walk. We then manually checked how many of the samples could be found in the Google index [2] (using the `inurl:` query option). We found that 23% the pages originating from the hard focus random walk and 28% of those originating from the trial-and-error random walk were not in the Google index. (We note that Google may deliberately discard pages from its index, e.g. if

they are duplicates of other pages.) This indicates that our random walk, even though running for a relatively short amount of time, was able to discover a variety of new pages related to the focus topic. We could not carry out more extensive experiments because most search engines do not allow automatic queries. Nevertheless, the above manual experimentation demonstrates that focused sampling may be a powerful tool for objective analysis and comparison of search engines.

# 6 Conclusions

Topical web statistics is crucial for generating opinion poll about products, market intelligence, tracking social networks, etc. Furthermore, in many scenarios, timely reporting of such statistics is a requirement. In this paper we described an algorithm that can uniformly sample web pages on a user-supplied topic. The algorithm is a random walk sampling algorithm that uses a classifier at each step to decide whether or not an outlink web page is on-topic. In our experiments we created two random walk algorithms: the hard focus algorithm and the soft focus algorithm. The hard focus algorithm does not allow an off-topic page to be selected. However, the soft focus algorithm allows the random walk to go to off-topic pages (within a particular distance). We experimented with two classifiers. The first classifier is a simple text classifier that classifies a page as on-topic if it contains the topic word. The second classifier is a decision tree classifier. We used the web pages in a local copy of a large crawl of the web as the baseline. Finally, we studied several statistical properties of web pages collected by each focused sampling algorithm. Our results indicate that the properties of the focused sample are similar to that of the baseline corpus.

# Acknowledgements

We thank Bruce Baumgart, Byron Dom, Nadav Eiron, David Gibson, Daniel Gruhl, Kevin McCurley, and Huaiyu Zhu for their help at various stages of this work.

# References

[1] AltaVista. `http://www.altavista.com`.

[2] Google. `http://www.google.com`.

[3] D. Aldous. On the Markov chain simulation method for uniform combinatorial distributed and simulated annealing. *Probability in the Engineering and Informational Sciences*, 1:33–46, 1987.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[5] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz. Approximating aggregate queries about web pages via random walks. In *Proceedings of 26th International Conference on Very Large Data Bases*, pages 535–544. Morgan Kaufmann, 2000.

[6] K. Bharat and A. Broder. A technique for measuring the relative size and overlap of public Web search engines. In *Proceedings of the 7th International World Wide Web Conference (WWW7)*, pages 379–388, April 1998.

[7] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference (WWW1998)*, pages 107–117, Brisbane, Australia, 1998.

[8] A. Broder and M. R. Henzinger. Algorithmic aspects of information retrieval on the web. In M. G. C. R. J. Abello, P. M. Pardalos, editor, *Handbook of Massive Data Sets*. Kluwer Academic Publishers, Boston, 2001.

[9] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web: experiments and models. In *Proceedings of the 9th International World Wide Web Conference (WWW9)*, May 2000.

[10] A. Broder, S. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference (WWW2000)*, pages 309–320, Amsterdam, The Netherlands, 2000.

[11] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kaufmann Publishers, 2002.

[12] S. Chakrabarti, M. Joshi, K. Punera, and D. Pennock. The structure of broad topics on the web. In *Proceedings of the 11th International World Wide Web Conference*. ACM Press, 2002.

[13] S. Chakrabarti, S. Roy, and M. V. Soundalgekar. Fast and accurate text classification via multiple linear discriminant projections. *The VLDB Journal*, 2003.

[14] S. Chakrabarti, M. van den Berg, and B. Dom. Distributed hypertext resource discovery through examples. In *Proceedings of the 25th International Conference on Very Large Databases (VLDB)*, pages 375–386, 1999.

[15] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: A new approach to topic-specific web resource discovery. In *Proceedings of the 8th International World Wide Web Conference (WWW8)*, pages 1623–1640, Toronto, Canada, 1999.

[16] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30:161–172, 1998.

[17] B. D. Davison. Topical locality in the web. In *Research and Development in Information Retrieval (SIGIR)*, pages 272–279, 2000.

[18] M. Diligenti, F. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused crawling using context graphs. In *Proceedings of 26th International Conference on Very Large Data Bases*, Cairo, Egypt, 2000.

[19] S. Dill, R. Kumar, K. S. Mccurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins. Self-similarity in Web. *ACM Transactions on Internet Technology*, 2:205–223, 2002. Online: `http://www.almaden.ibm.com/cs/k53/fractal.ps`.

[20] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, NY, 1973.

[21] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[22] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, MA, 1990.

[23] D. Gilman. A Chernoff bound for random walks on expander graphs. *SIAM J. on Computing*, 27(4):1203–1220, 1998.

[24] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najrok. Measuring index quality using random walks on the Web. In *Proceedings of the 8th International World Wide Web Conference (WWW8)*, pages 213–225, May 1999.

[25] M. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najrok. On Near-Uniform URL Sampling. In *Proceedings of the 9th International World Wide Web Conference (WWW9)*, pages 295–308, May 2000.

[26] A. K. Jain, P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[27] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[28] M. Jerum and A. Sinclair. Conductance and rapid mixiing of property for markov chains: the approximation of the permanent resolved. In *Proceedings of the Symposium on Theory of Computer Science*, pages 235–244, 1998.

[29] N. Kahale. Large deviation for Markov chains. *Combinatorics, Probability and Computing*, 6:465–474, 1997.

[30] T. Kanungo, C. H. Lee, and R. Bradford. What fraction of images on the Web contain text? In *Proceedings of Web Document Analysis*, 2001. Online: `http://www.csc.liv.ac.uk/ wda2001/ Papers/27_kanungo_wda2001.pdf`.

[31] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. In *Proceedings of the 8th International World Wide Web Conference (WWW1999)*, pages 1481–1493, Toronto, Canada, 1999.

[32] F. Menczer. Links tell us about lexical and semantic Web content. Technical Report cs.IR/0108004, Computer Science Department, Univ. of Iowa, 2001. Online: `http://arxiv.org/abs/cs.IR/0108004`.

[33] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[34] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Computer Science Department, Stanford University, 1998.

[35] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Wetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[36] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1992.

[37] J. Rennie and A. McCallum. Using reinforcement learning to spider the web efficiently. In *Proceedings of International Conference on Machine Learning*, 1999. Online: `http://www.cs.cmu.edu/ mccallum/papers/rlspider-icml99s.ps.gz`.

[38] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[39] P. Rusmevichientong, D. Pennock, S. Lawrence, and C. L. Giles. Methods for sampling pages uniformly from the World Wide Web. In *Proceedings of AAAI Fall Symposium on Using Uncertainity Within Computation*, Cape Cod, MA, 2001.