

Efficient Identification of Starters and Followers in Social Media

Michael Mathioudakis
Department of Computer Science
University of Toronto
mathiou@cs.toronto.edu

Nick Koudas
Department of Computer Science
University of Toronto
koudas@cs.toronto.edu

ABSTRACT

Activity and user engagement in social media such as web logs, wikis, online forums or social networks has been increasing at unprecedented rates. In relation to social behavior in various human activities, user activity in social media indicates the existence of individuals that consistently drive or stimulate ‘discussions’ in the online world. Such individuals are considered as ‘starters’ of online discussions in contrast with ‘followers’ that primarily engage in discussions and follow them.

In this paper, we formalize notions of ‘starters’ and ‘followers’ in social media. Motivated by the challenging size of the available information related to online social behavior, we focus on the development of random sampling approaches allowing us to achieve significant efficiency while identifying starters and followers. In our experimental section we utilize BlogScope, our social media warehousing platform under development at the University of Toronto. We demonstrate the scalability and accuracy of our sampling approaches using real data establishing the practical utility of our techniques in a real social media warehousing environment.

1. INTRODUCTION

Activity and user engagement in social media such as blogs (hosted in e.g., Blogger, Wordpress, LiveSpace), microblogging services (e.g., Twitter, Jaiku), wikis (e.g., Wikipedia), social networks (e.g., Facebook, MySpace, Friendster) and online discussion forums has been increasing at unprecedented rates. Indicative of the participation in social media, are the 200 million user profiles existing in MySpace and Facebook, the more than 50M active known web logs, the millions of users on Twitter, etc. Millions of individuals engage in online interactions on a daily basis reading and commenting on each others’ posts as well as exchanging ideas and thoughts.

In several online domains, activity is primarily attributed to a fraction of the individuals participating. In the old world of Usenet a study [9] identified only a fraction of ap-

proximately 3% of users answering questions in Usenet forums with the rest asking questions or engaging after the answer has been posted. Recent studies on Yahoo Answers [3] depict similar trends. On microblogging sites such as Twitter a few individuals have a very large number of ‘followers’ (users that get notified immediately when the individual posts or ‘twits’). The number of such followers satisfies a power law. Similar trends and power law phenomena arise in the case of social networks considering the number of friends or social connections [17].

In the case of weblogs (blogs) such phenomena arise when one considers information regarding readership of certain blogs or aggregate comments to posts on a blog. However in the case of blogs, user engagement takes more interesting forms. When a blogger publishes a post p , several other bloggers will read it. In addition though, blogging being a social process, some other bloggers will engage with that post, by publishing a post themselves that extends, criticizes, comments etc, on p and at the same time *link* to it. Such linking activity is indicative of a social engagement process that evolves as a function of time. In contrast, readership information can only be obtained for blogs that are available via popular feed services (e.g., feedburner), while commenting activity on blog posts is usually either anonymous or difficult to be linked back to the individual posting the comment. As a result, post linking activity in the blogosphere provides the best means to obtain an accurate picture for blogger engagement and reveals a lot of social activity among bloggers.

As is the case in various activities involving humans, some individuals primarily act as instigators or ‘starters’ of online discussions and some others primarily follow (are ‘followers’ of) such discussions and engage in them. Intuitively, we expect that a blogger who, over a significant period of time, primarily generates posts that others link (as opposed to primarily generating posts that link to other posts) will be considered a starter of discussions. In a similar fashion, a blogger that primarily links to other blog posts over a period of time can be regarded as a ‘follower’ of discussions. Consider for example bloggers¹ b_1 and b_2 in figure 1. Both bloggers publish posts that attract linking activity – and in fact both bloggers have attracted the same total number of inlinks to their posts. However, notice that most posts published on blog b_1 contain links to posts generated by other bloggers, indicating that b_1 was not the instigator or

¹We will assume for simplicity that a blogger b is associated with a single blog and we will use the same notation to refer to both.

‘starter’ of the related discussion. On the other hand, most of the posts published on blog b_2 do not link to posts generated by others, suggesting that b_2 was indeed the ‘starter’ of a discussion in which other bloggers engaged by creating their own posts and linking back to the posts of blogger b_2 . Our work focuses on identifying ‘starters’ like b_2 . Identifying such individuals is a task of extremely high value to advertisers, since in the online world, discussion starters act as ‘sources’ for the spread of messages.

As it is clear from the example in the previous paragraph, we would not be able to distinguish b_2 from b_1 as a ‘starter’ by simply counting the number of inlinks the related blogs have attracted. Instead, we should also take into account the number of outlinks from b_1 and b_2 to other blogs. A simple way to do this, which is also the approach we follow in this paper, is to compute the *difference*

$$d(b) = \#inlinks(b) - \#outlinks(b)$$

between the number of inlinks and outlinks related to a blog b and consider as ‘starters’ blogs b for which $d(b)$ is high (similarly, ‘followers’ are blogs with very small value of $d(b)$).

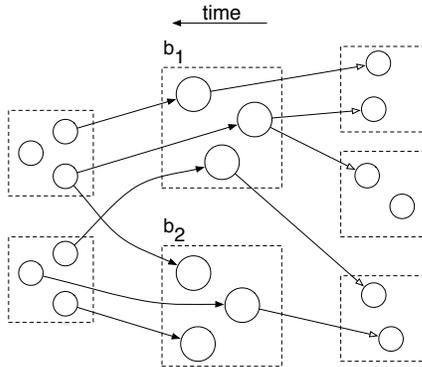


Figure 1: An example for ‘starters’ through the case of two bloggers b_1 and b_2 . In the figure, circles correspond to posts and directed edges correspond to links from one post to another. Posts from the same blog appear grouped together. The fact that blogger b_2 attracts the same number of inlinks with b_1 but creates smaller number of outlinks, indicates that blogger b_2 behaves more as a ‘starter’ of discussions than blogger b_1 .

At the University of Toronto, we have been building the infrastructure to collect in real time information published online by individuals, in the context of the BlogScope project ([5, 1]). Presently, the system tracks the Blogosphere (over 30M active blogs), Wikipedia, microblogging sites and news sources. We crawl the social media space and in real time collect, clean and aggregate millions of posts. Such posts are warehoused and form a text repository of the social media space as it evolves as a function of time. Considering only blog posts, our platform warehouses more than 300M posts. This wealth of information can be analyzed and mined in order to yield insights regarding online user activities. Given the volume of information involved, our techniques should be highly efficient and scalable. In this work, we develop effective sampling techniques that can significantly aid the task at hand and offer graceful tradeoffs between speed of execution and accuracy.

In particular, in this paper we make the following contributions:

- We formalize the notions of ‘starters’ and ‘followers’ in the blogosphere.
- We derive deterministic early-stopping conditions in the form of systems of linear inequalities. Such conditions, allow us to terminate early the computation of top k ‘starters’ or ‘followers’ while guaranteeing 100% accuracy for our results.
- We derive, subsequently, probabilistic early stopping conditions in order to achieve much *faster* identification of starters and followers with *accuracy guarantees*.
- We consider and analyze different random sampling approaches, depending on assumptions regarding our knowledge of the distribution of blog degrees (number of links).
- We develop a novel random walk based approach that results to few disk accesses when executed and subsequently yields an efficient sampling process. The approach adapts theoretical results from random-walk theory ([11]) to our setting.
- We demonstrate the scalability and accuracy of our approach using large amounts of real data, establishing the practical utility of our techniques in a real blog tracking environment.

The rest of the paper is organized as follows. In section 2, we describe previous work related to the problem we study in this paper. In section 3 we provide a formal definition of ‘starters’ and ‘followers’ and we introduce our notation and formal abstraction of the problem. Subsequently, in section 4, we discuss the efficiency issues that arise when one computes the sets of ‘starters’ and ‘followers’ in a real blog-tracking environment and provide probabilistic early-stopping conditions that allow to trade efficiency with accuracy. Probabilistic early stopping conditions depend on assumptions of uniform random sampling. We discuss in section 5 how this assumption can be satisfied and propose a random walk approach in section 6. In section 7, we provide experimental results that demonstrate the scalability and accuracy of our approach and conclude in section 8.

2. RELATED WORK

Information diffusion, namely the flow of information in social media has attracted a lot of research attention. In [13], [2] the authors model and study experimentally several aspects of information propagation in the Blogosphere, utilizing either linking activity between bloggers ([2]) or topic evolution ([13]). In [18], the authors study experimentally a large blog post dataset and demonstrate linking activity patterns.

In [12],[3], authors study social media activity in a more specialized context. In particular, in [12] the evolution and structure of discussions in Slashdot, a well known tech-news reporting web site, is analyzed. In [3], the authors focus on Yahoo! Answers, an online community question/answer portal and build a content analysis framework identifying high quality postings.

As it has been observed in many online domains (see [17] for example), a large fraction of activity is driven by a small

number of individuals. Such individuals, depending on the domain under study, are usually modeled as ‘leaders’ or ‘influential’. [21] identifies opinion leaders in the Blogosphere, by utilizing linking information and a modified PageRank scoring of graph nodes. However, the technique provided is static, time-consuming and not query-driven.

[8] formalized a notion of influence in order to model individuals that play a central role in information spread inside networks. The proposed model is interesting from an algorithmic / theoretical perspective, but costly to be applied in practice. Identification of individuals with maximal influence has been treated in [16].

Random walks have been utilized as a tool for random sampling on web graphs in [19],[6] and [14]. In particular, random walks have been used as a means to achieve approximately uniform sampling of *nodes* in a web graph, in order to estimate distributional properties – such as the fraction of web pages that belonged to a particular domain. Theoretical results related to the use of random walks for approximate counting were established in [4],[20] and [11]. We also employ random walks in our work, however we utilize it on the *edges* of web graphs rather than on their nodes, for reasons we explain in detail in sections 4-6.

3. PROBLEM FORMULATION

The blogosphere consists of a collection of bloggers and the set of their web logs (blogs). Blog search engines like BlogScope accept user-defined queries and return blog posts ranked according to various criteria including recency, relevance to the query, etc. Queries in the blogosphere, apart from specifying a set of keywords that should or should not be contained in the returned posts, often also have a temporal scope T , expressed as a multiple of a time unit (usually days). For example a query q^T would request all blog posts generated in the last T days that contain the keywords specified by q . Denote by P_q^T the result set of q^T in the temporal window T and let B be the set of blogs in P_q^T . We denote by P^b the set of all posts in P_q^T from blog b .

Although in principle the collection of blogs and posts does not have to be restricted in the result set P_q^T of q^T (we can conduct the same analysis on the entire collection of blogs and posts), we choose to identify starters and followers in a query focused manner. The reason is that in this way we can provide a more specific focus to our analysis by ensuring that all results will surely contain q . Note that q can be as general or as specific we choose. For the rest of the paper, set P_q^T will be denoted by P , implicitly assuming that it corresponds to a specific query q^T .

Moreover, links are often used by bloggers in their posts in order to refer to an interesting/related post created by another blogger. Given a particular post p , systems such as BlogScope, utilizing generic protocols implement the necessary mechanisms to both extract the links used by post p to refer to other posts and retrieve links used by other posts referring to post p^2 . A link l that appears in a post $p^{b_i} \in P^{b_i}$ of blog $b_i \in B$ and points to a post $p^{b_j} \in P^{b_j}$ will be denoted by an ordered pair $l = (p^{b_i}, p^{b_j})$ and all the links contained in a post p^b of a blog b will be denoted as L_{p^b} .

²This is the notion of trackbacks. In BlogScope since we warehouse all posts, each post has a unique identifier and we have a very efficient mechanism to retrieve trackbacks to a post given its unique post identifier.

Notation	
P	The set of all posts in the query result set
B	The set of blogs in a query result set P
p^b	The set of posts in P coming from blog B
L	The set of all links between posts in P
G	A graph used as an abstract representation of P and the links L (nodes represent posts, edges represent links between posts)
V	The node set of G
E	The edge set of G
MC	A Markov chain used to simulate a random walk on edges E

Figure 2: Explanation of notation used in the paper

We model such a setting using a directed graph G , with the posts $p \in P$ as the nodes of the graph and the links $l \in L = \bigcup_{b \in B} \bigcup_{p^b} L_{p^b}$ between blog posts as its edges. The situation is depicted in figure 3, where we show posts $p \in P$ that appear in a query result set P as nodes, the links l among posts as edges and where nodes corresponding to posts from the same blog have been grouped together.

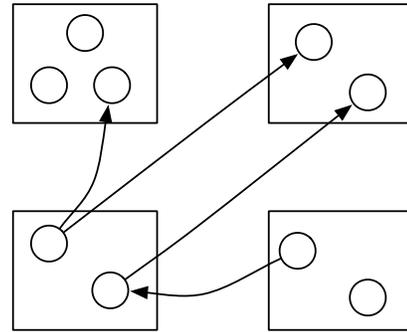


Figure 3: Graph G . Nodes (circles in the figure) correspond to posts $p \in P$ and directed edges (arrows) correspond to links $l \in L$ between posts. Posts from the same blog appear grouped together.

We consider as ‘starters’ bloggers that primarily generate posts that are linked by others while they seldom respond and link to other blogs. Similarly, we consider as ‘followers’ those that most often create posts that comment on and link to posts generated by other users, while they do not generate significant volume of posting and linking activity by other bloggers. Below, we provide the formal definitions for starters and followers, parametrized by a number k , together with some auxiliary definitions.

DEFINITION 1. The *in-degree* $inDeg_G(n_p)$ of a node n_p in graph G is defined as the number of its incoming edges.

$$inDeg_G(n_p) = \#\{l | l = (n'_p, n_p), n'_p \in P\}$$

DEFINITION 2. The *out-degree* $outDeg_G(n_p)$ of a node n_p in graph G is defined as the number of its out-coming edges.

$$outDeg_G(n_p) = \#\{l | l = (n_p, n'_p), n'_p \in P\}$$

DEFINITION 3. The **degree** $deg_G(n_p)$ of a node n_p in graph G is defined as the difference between its in-degree and out-degree.

$$deg_G(n_p) = inDeg_G(n_p) - outDeg_G(n_p)$$

Given a query result set P and its associated graph G , let p^b be a post of blog $b \in B$, n_{p^b} be the associated node in graph G and $sumDeg_G(b) = \sum_{p^b \in P^b} deg_G(n_{p^b})$ be the sum of degrees of all nodes n_{p^b} that correspond to posts in the result set from blog b .

DEFINITION 4. Given a query result set P and its associated graph G , the set S_k of the top k ‘**starters**’ among a set of blogs B is defined as the set of k blogs with maximum $sumDeg_G(b)$ ³.

DEFINITION 5. Given a query result set P and its associated graph G , the set F_k of the top k ‘**followers**’ among a set of blogs B is defined as the set of k blogs with minimum $sumDeg_G(b)$.

For the remainder of the paper, we refer to posts belonging to a pre-specified query result set P . Moreover we utilize the abstraction of graph G to implicitly refer to the real blog/post setting.

4. ANALYTICS

Given a query q^T on a system warehousing and searching blogs (like BlogScope), calculating ‘starters’ S_k or ‘followers’ F_k has a straightforward brute force solution. One can retrieve the entire set of posts $P = P_q^T$ that constitute the result set of query q , from the graph G as described in section 3, progressively computing $sumDeg(b)$ for all blogs $b \in B$ in the result set P and finally return the k ones with the largest value for $sumDeg(b)$. Commonly, retrieval of the post result set in the answer of q^T is provided through an iterator interface. Each call to the interface returns the next post (posts are ordered according to some ranking function). Since systems like BlogScope warehouse posts, a relational infrastructure may be utilized as a storage medium. Query searches against the data repository warehousing posts may return thousands of results (eg. posts). Each post is text (in html format) and may be typically several Kbytes in size. Retrieving all posts in a large result set incurs large overheads due to disk accesses to the post database, resulting in poor efficiency while computing S_k or F_k . Evidently the amount of time required to retrieve all posts in the result set may be high. Keyword queries against the post warehouse are issued utilizing inverted indices. A query is evaluated utilizing the inverted index infrastructure which in turn provides a list of unique post identifiers satisfying the query. In BlogScope we utilize a relational infrastructure to store posts. Thus, essentially we have access to the posts through an iterator interface, but also we can obtain random access to a post utilizing its post identifier (returned by the inverted index infrastructure).

We wish to derive a trade-off between the accuracy of the sets S_k , F_k and the number of posts retrieved from the result

³It is straightforward to extend the definition of ‘starters’ with the additional requirement that $sumDeg_G(b)$ of a ‘starter’ b exceeds a minimum value, without affecting the applicability of our methods - similarly for the definition of ‘followers’.

set P . We proceed towards this end in three steps. First, we derive deterministic early stopping conditions, i.e. conditions that would enable to stop the calculation of sets S_k and F_k early, while computing sets S_k , F_k precisely. Unfortunately, such deterministic conditions do not provide significant efficiency gains in practice. For this reason, we subsequently derive probabilistic early-stopping conditions, under the assumption of performing uniform random sampling on the edges of graph G . Finally, we highlight the difficulties in conducting such a random sampling procedure and detail our random walk solution in sections 5, 6.

For the rest of the paper, we detail primarily the computation of set S_k ; the same approach can be directly applied to compute F_k .

4.1 Deterministic early-stopping conditions

Let $V_b = \{n_{p^b} | p^b \in P^b\}$ be the set of all nodes n_{p^b} that correspond to posts from blog b , $V = \bigcup_{b \in B} V_b$ be the set of all nodes and E be the set of all directed edges $e = (n^{b_1}, n^{b_2})$. In order to determine the set S_k of starters, it is enough to traverse the set E of edges and calculate the degrees $deg(n)$ for all nodes $n \in V$. Our goal in this subsection is to derive conditions that will allow to avoid traversal of the entire set E and which, when satisfied, will guarantee that we obtain the exact result (100% accuracy). Traversal of set E can be achieved by visiting one by one all posts p in the query result set P and for each one obtaining and traversing its out-links.

Suppose that by traversing set E , we enumerate a subset E' of E . Let $deg_{E'}(n)$ be the degree of a node $n \in V$ taking into account only the directed edges in E' ; similarly let $S_k(E')$ be the set of k ‘starters’, calculated based on the subset $E' \subseteq E$ of edges. We will refer to $deg_{E'}(n)$ and $S_k(E')$ as the ‘current’ degrees and result set, respectively, in contrast with $deg_E(n)$ and $S_k(E) = S_k$, the ‘exact’ or ‘actual’ degrees and result set. The following observation holds.

OBSERVATION 1. If $S_k(E') \neq S_k(E) = S_k$, then there exists a pair of nodes (n, n') , with $n \in S_k(E')$ and $n' \notin S_k(E')$ such that $deg_E(n) < deg_E(n')$. \square

This observation states that if all nodes n' not belonging to the current result set $S_k(E')$, have actual degrees $deg_E(n')$ smaller than the actual degrees $deg_E(n)$ of nodes in $S_k(E')$, then they cannot belong to the actual set S_k of the top starters. We utilize this observation to derive our deterministic conditions in theorem 1.

THEOREM 1. Assume we have observed $E' \subseteq E$ of the edges and that each node $n \in V$ has at most out_i out-going and in_i in-coming edges in the edges $(E - E')$ currently unobserved. For each node n in the current result set $S_k(E')$ we construct a set of linear inequalities with M^2 variables x_{ij} , $1 \leq i, j \leq M = |B|$:

$$0 \leq \sum_j x_{ij} \leq out_i, \text{ for all } i, \quad 0 \leq \sum_j x_{ji} \leq in_i, \text{ for all } i,$$

$$deg_{E'}(n) + \sum_i x_{it} - \sum_j x_{tj} < deg_{E'}(n') + \sum_i x_{ib} - \sum_j x_{bj},$$

for all $n' \notin S_k(E')$

If none of the k sets of linear inequalities is feasible for the M^2 variables x_{ij} , then $S_k(E') = S_k(E) = S_k$. \square

In the theorem above, each variable x_{ij} can be seen as expressing the number of directed edges from node n_i to

node n_j in the currently unobserved subset $(E - E') \subseteq E$ of edges. Conditions

$$0 \leq \sum_j x_{ij} \leq out_i \text{ and } 0 \leq \sum_j x_{ji} \leq in_i$$

express the restrictions imposed on the number of out-going and in-coming edges per node. The last condition

$$deg_{E'}(n) + \sum_i x_{it} - \sum_j x_{tj} < deg_{E'}(n') + \sum_i x_{ib} - \sum_j x_{bj}$$

is satisfied when edges in $(E - E')$ are observed such that a node n' that is not in the current result set $S_k(E')$ finally surpasses in degree a node currently in $S_k(E')$. Therefore, it is not possible for the current result set $S_k(E')$ to change by observing the edges in $(E - E')$ if and only if none of the k systems of linear inequalities is feasible.

To utilize the deterministic early stopping conditions of Theorem 1, one needs to have estimates for the number of blogs M in a result set as well as for the upper-bounding restrictions in_i and out_i for out-going and in-coming edges per node. Upper bounds in_i and out_i can be derived in various ways. For example, in a blog tracking system, one might set a maximum acceptable number of links for a single blog during a suitable time granularity; otherwise the blog could be classified as ‘spam’. This kind of restrictions in the number of links for blogs can be directly translated to restrictions on in_i and out_i used in theorem 1.

Although in principle one may impose such restrictions, in practice such deterministic early-stopping rules like the one described in theorem 1 are loose, due to the large number of variables x_{ij} . Most importantly it is not easy to obtain a good estimation of M and tight bounds for in_i , out_i . As a result, the inequality system obtained in theorem 1 will have large domains and will be easily feasible. Choosing to use such early-stopping rules, will result in actually traversing almost the entire set E of edges before stopping and in no significant efficiency gain (see section 7).

For this reason, we relax our requirement for exact results and seek a trade-off between efficiency and accuracy with probabilistic guarantees.

4.2 Probabilistic early-stopping conditions

We develop a framework of probabilistic early stopping conditions that allow us to trade efficiency with accuracy while calculating sets S_k and F_k . Once again, we traverse set E and aim to return a set of starters $S_k(E')$ after observing $E' \subseteq E$ of edges. In order to obtain probabilistic accuracy guarantees for any such set $S_k(E')$, we make the following assumption.

ASSUMPTION 1. *The order in which the set E of edges is observed, is chosen uniformly at random.* \square

Thus, while traversing set E , the next edge is always selected without replacement uniformly at random from the remaining ones. We discuss more the implications and feasibility of assumption 1 in section 5.

Assume that only a (uniformly random) subset $E' \subseteq E$ of all edges has been observed. We calculate a set $S_k(E')$ of nodes as a current set of ‘starters’. Making use of observation 1, we turn our attention to the probability that there is a pair of nodes $n \in S_k(E')$ and $n' \notin S_k(E')$ such that $deg_E(n) < deg_E(n')$. More specifically, we examine each

pair of nodes (n, n') with $n \in S_k(E')$ and $n' \notin S_k(E')$ and estimate the probability that $deg_E(n) < deg_E(n')$. More formally, estimate the conditional probability (conditioned on the set of edges $E' \subseteq E$ observed)

$$Pr(deg_E(n) - deg_E(n') < 0 \mid E') \quad (1)$$

for all pairs of nodes n, n' with $n \in S_k(E')$, $n' \notin S_k(E')$. If this probability is small enough (e.g. less than 10% or 5%) for all such pairs, one can return the current result set $S_k(E')$ as an answer with high confidence.

To derive bounds for the probability in formula 1, we use Hoeffding’s inequality [15].

OBSERVATION 2. *Let $E_{\{n_i, n_j\}} \subseteq E$ be the subset of directed edges that have either n_i or n_j or both as adjacent nodes. Similarly, let $E'_{\{n_i, n_j\}} \subseteq E'$ be the subset of the observed directed edges that have n_i or n_j as an adjacent node. If $E' \subseteq E$ is chosen uniformly at random among all subsets of E of size $|E'|$, then $E'_{\{n_i, n_j\}}$ is also a uniformly random subset of $E_{\{n_i, n_j\}}$ among all subsets of $E_{\{n_i, n_j\}}$ of size $|E'_{\{n_i, n_j\}}|$.* \square

THEOREM 2 (HOEFFDING’S INEQUALITY[15]). *Let population U consist of N values c_1, c_2, \dots, c_N , each belonging to the interval $[a, b]$, with a mean value of*

$$\mu = \frac{1}{N} \sum_{i=1}^N c_i.$$

Also, let $U' = \{X_1, X_2, \dots, X_m\} \subseteq U$ denote a random sample without replacement from U with average value $\bar{X} = (X_1 + X_2 + \dots + X_m)/m$. Then,

$$Pr(\bar{X} - \mu \geq t \mid U') \leq e^{-\frac{2mt^2}{(b-a)^2}}.$$

\square

We apply Hoeffding’s result in this setting, to estimate the probability in formula 1 for a fixed pair of nodes (n, n') with $n \in S_k(E')$ and $n' \notin S_k(E')$. Consider the set $E_{\{n, n'\}}$ of directed edges as the population U of theorem 2 and the observed edges $E'_{\{n, n'\}}$ as a sample U' of size $|E'_{\{n, n'\}}| = m$. Observation 2 defends that $E'_{\{n, n'\}}$ is a random sample of $E_{\{n, n'\}}$ among all its subsets of the same size $|E'_{\{n, n'\}}| = m$.

Since the actual difference in degrees $deg_E(n) - deg_E(n')$ of the two nodes is of interest (see formula 1), every edge $e_i \in E'_{\{n, n'\}}$ will correspond to a value $c_i \in \{-2, -1, 1, 2\} \subseteq [-2, 2]$ of the theorem that is added to the difference $deg_E(n) - deg_E(n')$. Notice that each single directed edge $e_i \in E'_{\{n, n'\}}$ among the edges observed can either increase/decrease the difference by 2 (when both nodes are adjacent to the edge) or increase/decrease the difference by 1 (when only one of the two nodes is adjacent to the edge). In addition, no edge $e \in E'$ other than those in $E'_{\{n, n'\}}$ affects the difference $deg_E(n) - deg_E(n')$ and therefore, we have that

$$\begin{aligned} deg(n) - deg(n') &= deg_{E_{\{n, n'\}}}(n) - deg_{E_{\{n, n'\}}}(n') = \\ &= \sum_{i=1}^{|V|} c_i = \mu|V|, \end{aligned} \quad (2)$$

where $deg_{E_{\{n, n'\}}}(n)$ stands for the degree of node n based only on the subset $E_{\{n, n'\}} \subseteq E$ of edges.

Also, for our case $\bar{X}_{(n,n')} = \bar{X}$ represents the average degree difference of nodes n and n' in our sample $E'_{\{n,n'\}}$.

$$\begin{aligned}\bar{X} = \bar{X}_{(n,n')} &= \frac{\text{deg}_{E'_{\{n,n'\}}}(n) - \text{deg}_{E'_{\{n,n'\}}}(n')}{m} = \\ &= \frac{\text{deg}_{E'}(n) - \text{deg}_{E'}(n')}{m}\end{aligned}$$

Finally, using theorem 2 calculate a lower bound to the probability of the event $\{\text{deg}_E(n) - \text{deg}_E(n') < 0 | E'\}$ which, by equation 2, is equivalent to the event $\{\mu < 0 | E'\}$. Applying theorem 2,

$$\begin{aligned}Pr(\text{deg}(n) - \text{deg}(n') < 0 | E') &= Pr(\mu < 0 | E') \leq \\ &\leq Pr(\mu \leq 0 | E') = Pr(\bar{X} - \mu \geq \bar{X}) \leq \\ &\leq e^{-\frac{2m\bar{X}^2}{(2-(-2))^2}} = 1 - e^{-\frac{m\bar{X}^2}{8}} = \\ &= e^{-\frac{(\text{deg}_{E'}(n) - \text{deg}_{E'}(n'))^2}{8m}}\end{aligned}\quad (3)$$

Equation 3 provides the intuitive result that the larger the size $m = |E'_{\{n,n'\}}|$ of the sampled edges incident to either one of the two nodes n, n' and the larger the average difference of degrees (\bar{X}) between two nodes based on it, the smaller is the probability that the ordering of the two nodes (in terms of their degrees) will be different for the whole population of edges E . More specifically, if the square of the degree difference between the two nodes n, n' increases faster than the size of the sample of edges incident to them, then the probability that the sign of the degree difference will change after traversing all edges decreases.

This result has important consequences for degree distributions that are significantly skewed and heavy tailed, as it is most often the case for web graphs. In such cases, when a large fraction of edges are incident to a small number of nodes, one expects a uniformly random sample of the edges to capture the skew in the distribution of degrees, even if the size of the sample is small relatively to the total number of edges. Consequently, one expects the actual ‘starters’ to appear in our current result set $S_k(E')$ after a relatively small number of sampled edges E' . Also, the larger the skew in the degree distribution, the sooner one obtains high confidence that the current result set $S_k(E')$ is accurate. In our experimental evaluation (Section 7), we validate the intuition described in this paragraph providing results on real datasets.

5. RANDOM SAMPLING OF EDGES

Our discussion on probabilistic early stopping conditions has assumed that the order in which one observes the directed edges of set E is chosen uniformly at random. However, such an assumption is not always easy to satisfy in our scenario. To obtain a uniformly random sample of the edges E requires information on the distribution of the edges E that is difficult to obtain.

In what follows, we begin by examining some scenarios under which uniformly random sampling would be possible (subsections 5.1 and 5.2) and explain in more detail the difficulties that arise in each one. Subsequently, we discuss uniform random sampling of nodes $n \in V$ (subsection 5.3)

and its drawbacks against random sampling of edges $e \in E$. In section 6, we describe our random walk approach as a solution to the problems with uniform random sampling.

5.1 Distribution of out-degrees among nodes is known

We consider the case that uniformly random sampling of edges can be satisfied. In particular, we assume that the distribution of out-degrees among the nodes $n \in V$ is known.

Given $[V] = (n_1, n_2, \dots, n_{|V|})$ as an ordered list of nodes $n \in V$, let $Out_E(n_i)$ denote the total number of out-going edges contained in nodes $\{n_1, n_2, \dots, n_i\}$ (i.e. $Out_E(n_i)$ is the sum of out-degrees $\sum_{j=1}^i outDeg_E(n_j)$). If this distribution of out degrees is known, then choose a uniform random order to traverse the edges E , as described in algorithm 1.

Algorithm 1 randomEdge1

Input:List of nodes $[V]$, Edges E , Out-degree distribution $Out_E(n)$
Output:An ordered list of edges $[E]$
while E is not empty **do**
 Choose uniformly at random an integer $coin \in (0, |E|]$
 Visit the first node $n_i \in [V]$ such that $coin \leq Out_E(n_i)$
 Pick uniformly at random one edge e among the outgoing edges of n_i
 Remove edge e from E and append it to $[E]$
end while

return $[E]$

Algorithm 1 provides a traversal of the edges in random order, since at every iteration each edge $e = (e_i, e_j)$ (belonging to the out-going edges of a node n_i) is chosen with the same probability

$$\frac{outDeg_E(n_i)}{|E|} \frac{1}{outDeg_E(n_i)} = \frac{1}{|E|}.$$

Obviously, knowing the exact distribution of out-degrees among nodes is a strong assumption. In practice, knowing such a distribution for posts in an arbitrary query result set would require retrieving all posts $p \in D$ and extracting the links contained in them, canceling the utility of random sampling on links (the directed edges E in our graph abstraction). For this reason, in the next subsection we describe a scenario based on a weaker assumption regarding the distribution of edges among nodes and which, when satisfied, would allow uniform random sampling on the edges E .

5.2 The maximum out-degree of a node is known

We consider the case in which we have knowledge of the maximum out-degree $\max_{n \in V} \{outDeg_E(n)\}$ of a node $n \in V$ – or more generally, an upper bound out_{max} . Algorithm 2 provides edges E in random order.

Algorithm 2 provides edges uniformly at random, since at every repetition, the probability that edge $e = (n_i, n_j)$ is chosen equals

$$\frac{1}{|V'|} \frac{outDeg_E(n_i)}{out_{max}} \frac{1}{outDeg_E(n_i)} = \frac{1}{|V'| \cdot out_{max}}$$

where V' is the set of nodes $n \in V$ with at list one out-going edge $e \in E$.⁴

⁴Notice that out-degrees $outDeg_E(n_i)$ and set V' are updated during the procedure of algorithm 2 as we remove

Algorithm 2 randomEdge2

Input:Set of nodes V , Edges E , out-degree upper bound out_{max}
Output:An ordered list of edges $[E]$
while E is not empty **do**
 Choose uniformly at random an integer $coin_1 \in (0, |V|]$
 Visit the node $n_{coin_1} \in V$
 Choose uniformly at random a real number $coin_2 \in [0, 1)$
 if $coin_2 < \frac{f_i}{f_{max}}$ **then**
 Pick uniformly at random one edge e among the out-going edges of node n_{coin_1}
 Remove edge e from E and append it to $[E]$
 end if
end while

return $[E]$

However, even if knowledge of an upper bound out_{max} on the out-degrees of the nodes in V is assumed there still exists the following drawback; for heavy-tailed distributions of edges among nodes, one would have to visit many nodes before actually picking one edge $e \in E$ for the sample. This could lead to visiting almost all nodes before obtaining our sample; this eliminates the advantage of the sampling process. More specifically, the probability of actually picking an edge in our sample in a single iteration of the procedure is expressed using the following sum over nodes that contain at least one out-going edge

$$r = \sum_{n \in V'} \frac{outDeg_E(n)}{|V'| \cdot out_{max}} = \frac{\sum_{n \in V'} outDeg_E(n)}{|V'| \cdot out_{max}} = \frac{|E|}{|V'| \cdot out_{max}}.$$

By a known result for Bernoulli trials, the expected number of iterations before deriving an edge for the sample is

$$\frac{1}{r} = |V'| \cdot \frac{out_{max}}{|E|}.$$

Thus, if $\frac{out_{max}}{|E|} \gg \frac{1}{|V'|}$, we'll need to visit several nodes before we actually select one edge $e \in E$.

5.3 Sampling nodes uniformly at random

We consider the use of random sampling on the nodes V , as shown in algorithm 3.

Algorithm 3 randomNodes

Input:Set of nodes V , Edges E
Output:An ordered list of edges $[E]$
while E is not empty **do**
 Choose uniformly at random an integer $coin_1 \in (0, |V|]$
 Visit the node $n_{coin_1} \in V$
 Pick all the out-going edges of n_{coin_1} , remove them from E and add them to $[E]$
end while

return $[E]$

This procedure (Algorithm 3) does not produce edges $e \in E$ sampled uniformly at random. Nonetheless, we consider edges e from E , since the selection of edges is without replacement.

this method because it leads to *unbiased* estimates regarding the number of directed edges from a node n_i to another node n_j , i.e., estimates for which the average value among all possible samples is equal to the true value of the quantity to be estimated. However, an unbiased estimate can suffer from big standard errors, especially for skewed, heavy-tailed distributions (see [7]) as in our case and can lead to poor accuracy, as we show experimentally in section 7.

6. THE RANDOM WALK APPROACH

Uniform random sampling on the set of edges E either requires strong assumptions (which not necessarily hold in practice) or eliminates any significant efficiency gains. For this reason, we resort to a *random walk* approach or *Markov chain simulation* method as an approximation to uniform random sampling. Theorem 3 below, provides the basic result related to Markov chain simulation [11].

THEOREM 3 (CONVERGENCE OF MARKOV CHAIN SIMULATION).

Let $G(V, E)$ be a non-directed connected graph, $s \in V$ be a node and $A \subseteq V$ be a subset of nodes in the graph. Associate with the graph G a Markov chain MC where transitions from a node to its neighbors happen uniformly at random. Consider the following procedure (the “Aldous procedure”).

Choose a positive integer k . Start the random walk at node s and simulate it for k steps (the “delay”). Let x_0 be the final state (node) of the walk. Choose another positive integer l . Starting from x_0 , continue the random walk for l more steps taking each subsequent point x_i , $1 \leq i \leq l$, as a sample point. Store in variable t_1 the number of sample points in A and let $t = t_1/l$.

Suppose we choose $k = \log(1/\pi(s))/\epsilon$ and $l = 20 \log(8/\delta)/(\epsilon\beta^2\pi(A)^2)$, where ϵ is the eigenvalue gap $\epsilon = 1 - \lambda_2$ of the transition matrix of MC (λ_2 is the second largest eigenvalue of the transition matrix). Then with probability at least $1 - \delta$,

$$\pi(A)(1 - \beta) \leq t \leq \pi(A)(1 + \beta),$$

where $\pi(A)$ is the probability that the Markov chain MC is at a node in A in steady state. □

It is well known ([10]) that if G is regular (i.e. all nodes have the same degree) and connected, then its stationary distribution is uniform. In this case, $\pi(A) = \frac{|A|}{|V|}$ and $\pi(s) = \frac{1}{|V|}$. According to theorem 3, when it is applied on a regular connected graph, we need $k = O(\log(|V|))$ steps in the random walk to remove any significant bias due to the starting node of the walk and that the accuracy of counting the size of any subset of nodes A is increasing exponentially with the number l of steps thereafter. Consequently, as the number of steps $(k + l)$ of a random walk on a regular graph tend to infinity, the distribution of the visited nodes tends to be uniform. Therefore, one can use a random walk approach on a graph for sampling nodes, with the sampling approaching uniform as the length of the walk increases.

We are interested in obtaining a uniform random sample of the edges E of graph G rather than its nodes V . For this reason we apply a careful transformation to graph G , obtaining a regular Markov chain MC on the edges E of G . We achieve this in two steps.

In the first step, from the directed graph $G(V, E)$ we obtain another graph $H(V, E_1)$, with $E \subseteq E_1$, that satisfies the following three properties: (1) is undirected, (2) is regular and (3) is connected. Property (1) is satisfied removing the direction from edges $e \in E$. In practice, this means that given a post p , the search engine can both extract and return the links contained in it and retrieve the links contained in other posts that point to post p . In systems like BlogScope the mechanisms necessary to perform these tasks are readily available. Property (2) is satisfied by adding self-loops to nodes of graph G . The process is depicted in figure 4. Finally, property 3 is satisfied by adding special undirected edges or *hops* between pairs of nodes that correspond to posts that are returned sequentially in the query result by the blog search engine. This is easily accomplished in our setting at the iterator interface level, while retrieving consecutive posts.

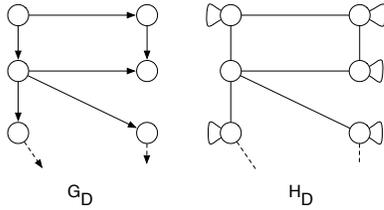


Figure 4: Graphs G and H . Graph H is produced from G by removing the direction of its edges E and adding self-loops to the nodes so that they have the same degree (e.g. a degree 4 for the example shown in the figure).

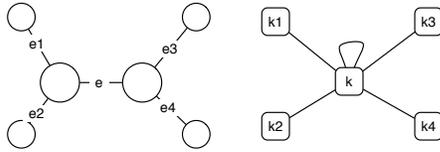


Figure 5: Graph H and Markov chain MC . State k of MC corresponds to undirected edge e of graph H , state k_1 corresponds to edge e_1 and so on. Two states in MC communicate through a transition only if the corresponding edges share a common adjacent node. In the figure, we show only transition from and to state k . Notice that k has self loops, since edge e trivially has a common adjacent node with itself.

In the second step, we obtain from H a Markov chain $MC(K, T)$ with the set K representing its states (its ‘nodes’) and the set T representing the possible transitions (its ‘edges’) from state to state. In addition, we allow a transition $t \in T$ between two states k_1 and $k_2 \in K$ if and only if there is an edge $e \in E_1$ between the associated nodes n_1 and $n_2 \in V$ in graph H . The construction of the Markov chain MC is depicted in figure 5.

OBSERVATION 3. *The Markov chain MC is regular.* \square

Based on this observation and on theorem 1, we can simulate a random walk on MC , constructed as outlined, in order to obtain a sample of its states K and therefore a sample

of edges E . This sample will be closer to uniform as we increase the length of our random walk on the Markov chain MC .

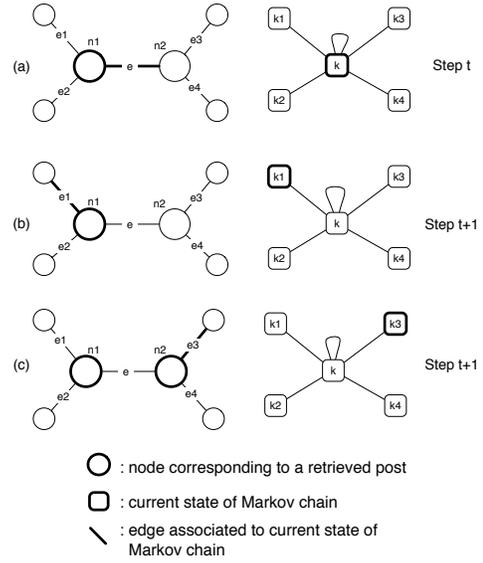


Figure 6: Random Walk. The figure demonstrates how our random walk moves from state to state of the Markov chain MC and from edge to edge in graph H .

The random walk on a real blog-tracking system like BlogScope is explained through figure 6, which demonstrates an instance of graph H and Markov chain MC ⁵. Suppose that Markov chain MC is at state k at step t of the random walk, or, in terms of graph H , at edge e . Edge e corresponds to a link $l \in L$ and presumably, it has been extracted (through the link extraction mechanisms mentioned earlier in the paper) by the post p_1 that corresponds to node n_1 . Therefore post p_1 has already been retrieved from our post database.

At step $(t+1)$, the random walk follows a transition to one of the states k_1, \dots, k_4 or a loop to current state k . In terms of graph H , the walk moves to one of the edges e_1, \dots, e_4 or stays at the current edge e . In general, since both nodes n_1 and n_2 have the same degree in graph H , the random walk moves with probability $\frac{1}{2}$ to an edge incident to each node n_1 or n_2 . If it moves to an edge incident to n_1 (Figure 6(b)), then the corresponding post p_1 has already been retrieved by our system and no access cost incurs. Otherwise (Figure 6(c)), post p_2 has to be retrieved in order for the random walk to move to one of its edges. In such case, a random access cost to the database incurs.

Section 4.2 provided bounds (Formula 3) related to the probability that one could stop early the computation of ‘starters’ S_k or ‘followers’ F_k by observing only a subset of the links in a result set. The bound of formula 3 cannot be used as a probability in this setting, since the random walk is equivalent to random sampling of edges only in the limit of infinite steps. We use it as a heuristic score, however, in order to obtain an intuition as to when to terminate the walk.

In fact, at every step of the random walk and for each pair

⁵For simplicity *hops* are not included in this example

of nodes⁶ (n, n') with $n \in S_k(E')$ and $n' \notin S_k(E')$, a score

$$score(n, n') = 1 - e^{-\frac{(deg_{E'}(n) - deg_{E'}(n'))^2}{8m}}$$

is computed, where $m = |E'_{\{n, n'\}}|$. The average value of this score over all pairs of nodes (n, n') with $n \in S_k(E')$ and $n' \notin S_k(E')$ is reported as our *confidence score*.

$$confScore = Avg_{(n, n')} \{score(n, n')\}$$

When *confScore* exceeds a predefined threshold (at the level, say, of 80% or 90%), the walk is terminated and the current set of ‘starters’ or ‘followers’ is returned as an answer.

7. EXPERIMENTS

Quantitative and qualitative experiments were performed to evaluate the scalability and practical utility of the random walk technique. We utilize BlogScope to conduct our experiments. BlogScope currently warehouses over 370M posts from active blogs for a total of over 3 TB of social media data.

In section 7.1 we present real examples of ‘starters’ in the blogosphere for several query result sets. Section 7.2 reports efficiency and accuracy results of the random walk (Section 6) and random posts (Section 5.3) approaches on real datasets of varying size and compares them with a straightforward computation of sets S_k and F_k .

Keywords	Temporal Scope
{obama, clinton, mccain}	[2007, Jan 1st - 2008, May 31st]
{gadgets, software}	[2007, Jan 1st - 2008, May 31st]
{hollywood}	[2007, Jan 1st - 2008, May 31st]
{yahoo, microsoft, google}	[2007, Jan 1st - 2008, May 31st]
{finance, economic investment, business administration, stock exchange}	[2007, Jan 1st - 2008, May 31st]

Figure 7: The queries used for the qualitative experiments. Issued to BlogScope, they return posts that contain at least one of the terms in the query – for example, ‘obama’ or ‘clinton’ or ‘mccain’ for the first one.

7.1 Qualitative Results

In this section we present typical results, identifying starters for several queries using BlogScope. We submit a query to BlogScope specifying a temporal interval and compute starters on the results returned. Typical queries used are shown in figure 7.

Rank	Blog URL	Degree
1	gatewaypundit.blogspot.com	806
2	digbysblog.blogspot.com	565
3	thirdstatesundayreview.blogspot.com	559
4	demconwatch.blogspot.com	539
5	shakespeareessister.blogspot.com	466

keywords: {obama, clinton, mccain}, T = [Jan 1st, 2007 - May 31st, 2008]

Figure 8: Top starters for the query {obama, clinton, mccain}.

⁶Notation follows that of section 4.

Rank	Blog URL	Degree
1	softwarecomplex.blogspot.com	195
2	googlesystem.blogspot.com	139
3	labnol.blogspot.com	138
4	googleblog.blogspot.com	98
5	ps3guru.blogspot.com	75

keywords: {gadgets, software}, T = [Jan 1st, 2007 - May 31st, 2008]

Figure 9: Top starters for the query {gadgets, software}.

Given a query, a graph G is created as described in section 3. For each such graph G , the top 5 ‘starters’ are obtained as shown in figures 8- 12 (the ‘degree’ column in the tables corresponds to the notion of degree defined in section 3).

Rank	Blog URL	Degree
1	unitedhollywood.blogspot.com	149
2	zlebs2.blogspot.com	103
3	storystructure.blogspot.com	60
4	hollywood-infotainmentindia.blogspot.com	46
5	kenlevine.blogspot.com	36

keywords: {hollywood}, T = [Jan 1st, 2007 - May 31st, 2008]

Figure 10: Top starters for the query {hollywood}.

Rank	Blog URL	Degree
1	googlesystem.blogspot.com	2506
2	googleblog.blogspot.com	1065
3	windowsup.blogspot.com	998
4	softwarecomplex.blogspot.com	519
5	labnol.blogspot.com	511

keywords: {yahoo, microsoft, google}, T = [Jan 1st, 2007 - May 31st, 2008]

Figure 11: Top starters for the query {yahoo, microsoft, google}.

Blog search engines, including BlogScope, contain various mechanisms to quantify the ‘authority’ of a blog. One such measure is the total number of in-links to the blog. One natural question that arises is whether the model proposed is practically equivalent to a model that would identify ‘starters’ based only on the in-degree of the blog and ‘followers’ only based on the out-degree (per definitions provided in section 3). As we show in the example of figure 13, these two models return very different sets of blogs in their top results. In figure 13, we present top starters identified for the query {obama, clinton, mccain} in the 3.5-months interval (July 1st 2007 - October 15th 2007). Although blogs like ‘thirdstatesunday.blogspot.com’ or ‘wwwmikeylikesit.blogspot.com’ (extremely popular blogs related to politics) had many in-links during that period of time, they also created many posts referring and linking to other blogs. As a result, they do not appear in the top 5 ‘starters’ – actually, both blogs ranked below the 20th position of ‘starters’. In-

Rank	Blog URL	Degree
1	businessportal.blogspot.com	305
2	everydayfinance.blogspot.com	177
3	brokersreports.blogspot.com	159
4	eubanking.blogspot.com	138
5	firefinance.blogspot.com	105

keywords: {finance, economic investment, ...},
T = [Jan 1st, 2007 - May 31st, 2008]

Figure 12: Top starters for the query {finance, economic investment, business administration, stock exchange}.

stead, in the top 5 ‘starters’ appear blogs like ‘gatewaypundit.blogspot.com’ or ‘likemariasaidpaz.blogspot.com’ that attracted much linking activity generating a lot of original posts (i.e. posts that did not refer or link to posts by other blogs). Noticeably, ‘thirdstatesunday.blogspot.com’, one of the blogs with most inlinks during that period, generated several posts that linked to original content (mainly on the US presidential race) generated by ‘likemariasaidpaz.blogspot.com’, one of the top ‘starters’. Similar results were obtained for other queries and other time intervals as well.

Rank	Top 5 Starters	Top Inlinks
1	thecommonills.blogspot.com	thecommonills.blogspot.com
2	gatewaypundit.blogspot.com	thirdstatesundayreview.blogspot.com
3	likemariasaidpaz.blogspot.com	wwwmikeylikesit.blogspot.com
4	althouse.blogspot.com	cedricsbigmix.blogspot.com
5	mcbridesmediamatters.blogspot.com	thedailyjot.blogspot.com

keywords: {obama, clinton, mccain} T = [2007, Jul 1st - 2007, Oct 15th]

Figure 13: Lists of top ‘starters’ and blogs with highest in-degree for query {obama, clinton, mccain} with temporal scope [July 1st 2007 - Oct 15th 2007].

7.2 Quantitative Experiments

This section aims to demonstrate the scalability and accuracy of the random walk approach (Section 6). A query q is issued to BlogScope for a temporal interval T ; the set of blog post urls in the result are retrieved through inverted indices and subsequently the actual set of results P as well as the links L between them are obtained.

Posts are retrieved via an iterator interface utilizing the list of blog post urls. Posts are also retrieved via random accesses. In BlogScope given a post url in the result set we can easily retrieve the actual post from the post database. Once a post is retrieved it is easy to identify all links from that post to other posts. In addition given a post url, we can easily, via standard interfaces supplied by BlogScope, retrieve all links from other posts that link to that post url.

Computing sets S_k and F_k in a straightforward manner (by accessing all posts $p \in P$, extracting the links L and calculating S_k or F_k on G) involves only retrieval of posts through the iterator interface.

In section 4.1, deterministic early stopping conditions were derived for the calculation of sets S_k and F_k . Such deterministic early stopping conditions require estimates for the number M of blogs in a result set as well as tight bounds for

Keywords	Temporal Scope	Number of Links L
{yahoo, microsoft, google}	[2007, Jan 1st - 2008, May 31st]	42842
{obama, clinton, mccain}	[2007, Jan 1st - 2008, May 31st]	22641
{gadgets, software}	[2007, Jan 1st - 2008, May 31st]	6629

Figure 14: Queries used for the quantitative results.

parameters out_i, in_i . We experimentally used deterministic early stopping conditions for cases that M was known and tight bounds for out_i, in_i were available and we found that there were no efficiency gains while computing sets S_k and F_k compared to a brute-force computation. For this reason, we do not report any results for this approach.

In summary, in the experiments that follow we compare three techniques to compute ‘starters’ and ‘followers’.

- **BruteForce:** This is the straightforward way to compute a set of ‘starters’ S_k or ‘followers’ F_k .
- **RandomWalk** This is the random walk approach, described in section 6.
- **RandomPosts** This method is depicted in algorithm 3 of section 5.3. Only random accesses to the post database occur during the execution of this method, since posts are retrieved in random order.

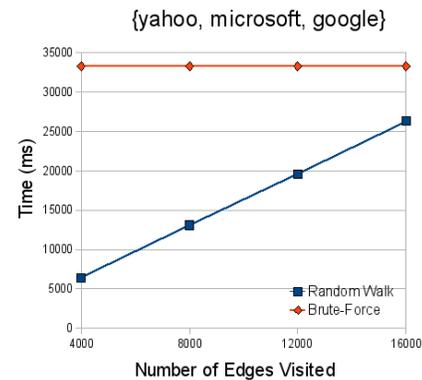


Figure 15: Time performance of BruteForce and RandomWalk for the {yahoo, microsoft, google} result set.

All techniques are compared in terms of both efficiency and accuracy.

- **Efficiency** is measured as the total time required.
- **Accuracy** for the **RandomWalk** and **RandomPosts** methods is measured as the percentage of the top 10 ‘starters’ identified, belonging also to the list of top 10 ‘starters’ calculated using **BruteForce**.

The techniques are evaluated on three result sets, obtained through BlogScope by issuing the queries shown in figure 14. In the figure, we also show the number of links $|L|$ in each result set - and equivalently the number of edges $|E|$ in graph G .

RandomWalk is executed for different walk lengths on each result set. Figures 15,17 and 19 present averages over 30 runs for the total time required. For every walk length,

average accuracy is reported in figures 16, 18 and 20. Moreover, for walks of the same length and for each repetition of `RandomWalk`, the `RandomPosts` technique is executed for the same time as `RandomWalk` (i.e. is allowed to pick randomly as many posts required in order to have the same execution time as that of `RandomWalk`). We report average accuracy of `RandomPosts` on the same graphs (Figures 16, 18 and 20).

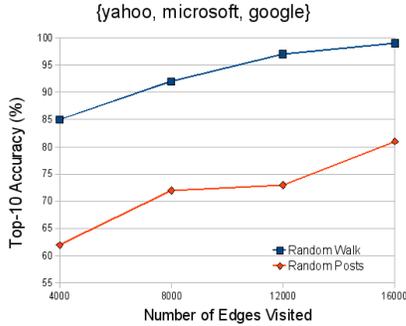


Figure 16: Accuracy comparison of `RandomWalk` and `RandomPosts` for `{yahoo, microsoft, google}` result set.

All the experiments demonstrate two facts for the `RandomWalk` technique. First, it exhibits good scalability, i.e. it provides good accuracy for walk lengths that are only a fraction of the total edge set size (E). For example for graph G produced by the query `{yahoo, microsoft, google}` (Figures 15,16), `RandomWalk` provides over 95% accuracy for a walk of length nearly one third ($1/3$) of the edge set size $|E|$. Secondly, we observe that `RandomWalk` exhibits consistently better accuracy than `RandomPosts`. For example, in figures 15,16 we observe that `RandomWalk` constantly outperforms `RandomPosts` by nearly a 20% accuracy margin, for the same total access cost.

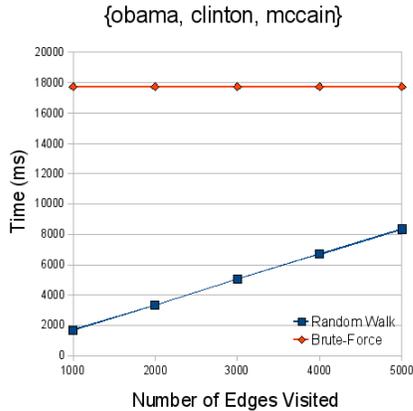


Figure 17: Time performance of `BruteForce` and `RandomWalk` for the `{obama, clinton, mccain}` result set.

Finally, in figure 21 and for the result set of query `{gadgets, software}`, we plot the confidence score `confScore` introduced at the end of section 6 and compare it with the accuracy of `RandomWalk` as a function of the random walk length. We note that the two curves behave in a similar way and are really close. Similar observations hold for the confidence scores of the result sets for queries `{obama, clin-`

`ton, mccain}` and `{yahoo, microsoft, google}` among others. Therefore, `confScore` could be used as heuristic real-time estimation of `RandomWalk`'s accuracy so as to terminate the walk when it exceeds an appropriately set threshold. Therefore `confScore` provides a heuristic way to stop the random walk early, by observing the values of `confScore` in real time as the walk takes place. As shown in the figure, accuracy at the 90% level is attainable.

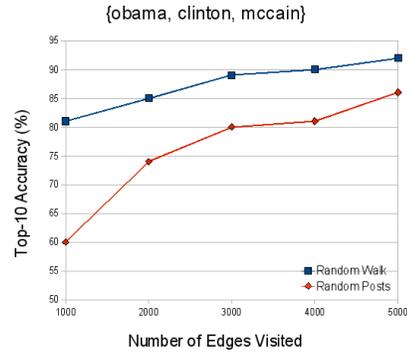


Figure 18: Accuracy comparison of `RandomWalk`, `RandomPosts` for the `{obama,clinton,mccain}` result set.

8. CONCLUSION

In this paper, we formalized notions of ‘starters’ and ‘followers’ in social media and focused on their efficient computation in real blog tracking systems. More specifically, we considered the usage of random sampling approaches and developed probabilistic early stopping conditions that allow to achieve fast identification of starters and followers with accuracy guarantees. Moreover, we developed a random walk based technique that results to few disk accesses when executed and therefore provides an efficient sampling process. Finally, in our experimental section we demonstrated the scalability and accuracy of our approach using the `BlogScope` search engine.

9. ACKNOWLEDGEMENTS

We wish to thank Nilesh Bansal for his insightful comments and valuable assistance.

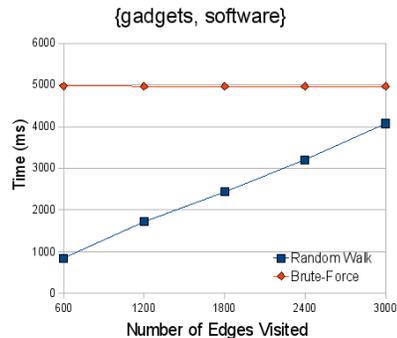


Figure 19: Time performance of `BruteForce` and `RandomWalk` for the `{gadgets, software}` result set.

10. REFERENCES

- [1] Nilesch Bansal and Nick Koudas, *BlogScope: A System for Online Analysis of High Volume Text Streams, WebDb*, 2007.
- [2] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 207–214, Washington, DC, USA, 2005. IEEE Computer Society.
- [3] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 183–194, New York, NY, USA, 2008. ACM.
- [4] D. Aldous. On the markov chain simulation method for uniform combinatorial distributions and simulated annealing. *Probability in the Engineering and Informational Sciences*, 1987.
- [5] N. Bansal, F. Chiang, N. Koudas, and F. W. Tompa. Seeking stable clusters in the blogosphere. In *VLDB*, pages 806–817, 2007.
- [6] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenphol, and D. Weitz. Approximating aggregate queries about web pages via random walks. In *VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases*, pages 535–544, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [7] W. Cochran. *Sampling Techniques*. John Wiley and Sons, 3rd edition, 1977.
- [8] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, New York, NY, USA, 2001. ACM.
- [9] D. Fisher, M. Smith, and H. T. Welser. You are who you talk to: Detecting roles in usenet newsgroups. In *HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Washington, DC, USA, 2006. IEEE Computer Society.
- [10] R. Gallager. *Discrete Stochastic Processes*. Springer, 1st edition, 1995.
- [11] D. Gillman. A chernoff bound for random walks on expander graphs. *SIAM J. Comput.*, 27(4):1203–1220, 1998.
- [12] V. Gómez, A. Kaltenbrunner, and V. López. Statistical analysis of the social network and discussion threads in slashdot. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 645–654, New York, NY, USA, 2008. ACM.
- [13] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 491–501, New York, NY, USA, 2004. ACM.
- [14] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform url sampling. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications networking*, pages 295–308, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.
- [15] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- [16] D. Kempe, J. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, New York, NY, USA, 2003. ACM.
- [17] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *KDD '08*.
- [18] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs, 2007.
- [19] P. Rusmevichientong, D. M. Pennock, S. Lawrence, and L. C. Giles. Methods for sampling pages uniformly from the world wide web. In *AAAI Fall Symposium on Using Uncertainty Within Computation*, pages 121–128, 2001.
- [20] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing markov chains. *Inf. Comput.*, 82(1):93–133, 1989.
- [21] X. Song, Y. Chi, K. Hino, and B. Tseng. Identifying opinion leaders in the blogosphere. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 971–974, New York, NY, USA, 2007. ACM.

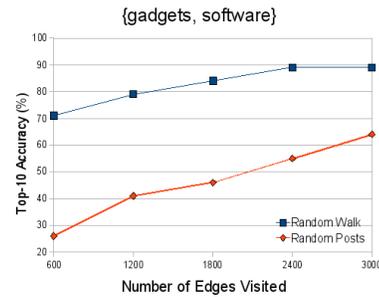


Figure 20: Accuracy comparison of RandomWalk and RandomPosts for the {gadgets, software} result set.

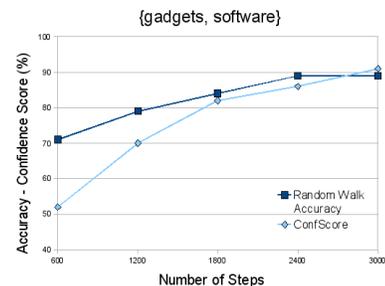


Figure 21: Comparison between Confidence Score and Accuracy for the {gadgets, software} result set.