Extracting and Ranking Viral Communities Using Seeds and Content Similarity

Hyun Chul Lee*
University of Toronto
Toronto, ON, Canada
leehyun@cs.toronto.edu

Allan Borodin†
University of Toronto
Toronto, ON, Canada
bor@cs.toronto.edu

Leslie Goldsmith
Affinity Systems
Mississauga, ON, Canada
lhg@affsys.com

ABSTRACT

We study the community extraction problem within the context of networks of blogs and forums. When starting from a small set of known seed nodes, we argue that the use of content information (beyond explicit link information) plays an essential role in the identification of the relevant community. Our approach lends itself to a new and insightful ranking scheme for members of the extracted community and an efficient algorithm for inflating/deflating the extracted community. Using a considerably large commercial data set of blog and forum sites, we provide experimental evidence to demonstrate the utility, efficiency, and stability of our methods.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: database applications—*Data mining*; I.5.3 [**Pattern Recognition**]: clustering—*Algorithms,Similarity measures*

General Terms

Algorithms, Experimentation, Performance

Keywords

Community, Extraction, Ranking, Similarity

1. INTRODUCTION

Recently, there has been substantial interest in the problem of discovering community structures from a web graph based primarily on the hyperlink structure of the graph [1, 9, 16, 18, 22, 27]. Most of the previous work defines the intuitive notion of a web community as a subgraph of a given web graph whose members are, in some sense, *more similar* to each other than to other, non-community members. The basic measure employed to represent such similarity (or dissimilarity) is the linkage relations among members of the given web graph. Although various research papers (e.g. [15, 3]) suggest ways to incorporate node and edge weights in the process, the utility of doing is still not fully understood. In this paper, we begin a more comprehensive approach to community extraction, as we are especially interested in a methodology for studying community structure and growth in the viral word-of-mouth segment of the web as exemplified by blogs and forums. In this context, hyperlinks are somewhat marginalized: Experience analyzing web content at Brandimensions[1] has shown that the interlinking structure of forums, blogs, and other community content sites is somewhat different from that typical on the more authoritative part of the web and, in particular, that graphs generated from such pages are relatively sparse. [2] Furthermore, where links do exist, community sites often link not to each other, but rather directly to the source of the underlying information. At the level of individual postings, we can never have bidirectional links as postings have inherent time stamps. We argue that successful community extraction schemes in the viral community context must adequately account for and properly weight both linkage *and* node information.

We propose three different problem settings. In the first (baseline) setting, we assume that the user provides a set of seed pages, some potentially classified as good (i.e. relevant) and some as bad (irrelevant) from which we want to extract the community. The community extraction problem then becomes one of discovering the pages that are most similar to the given good seed pages while being most dissimilar to the given bad ones. In the second setting, we additionally construct edge weights for the given web graph by taking into account both the page content and the linkage-based relations of pages. Content information is used as a factor in determining edge weights as well as "conservatively" creating some new edges based upon lexical similarity [3]. This approach has previously been shown to be effective for other web mining applications like the classification of hyper-linked document objects [2] and the ranking of forum pages [25]. Our goal here is to obtain a better understanding of the inherent community structure(s) being defined by going beyond the simple linkage-based relations of pages. Our view is that belonging to a community is a more refined concept than just some topic similarity. The approach we develop lends itself to reshaping through the inflating and deflating of communities (as advocated in [10]).

We are fortunate to have an extensive database for many communities including some carefully determined (by human experts) seed nodes. Hence, our first (base) and second settings reflect sit-

*Research supported by Brandimensions

†Research supported by Brandimensions and MITACS

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HT'08, June 19-21, 2008, Pittsburgh, Pennsylvania, USA.
Copyright 2008 ACM 978-1-59593-985-2/08/06 ...$5.00.

[1] http://www.brandimensions.com

[2] In our experimental data set of blogs and forums, we observed that the average number of hyperlinks per blog/forum was 2.1. This is considerably less than what is reported for standard web pages.

[3] We think of lexical similarity as a simple approximation for true semantic similarity. A better approximation can be obtained, for example, by using distributional clustering of words as advocated in Pereira *et al* [20].

uations where there exists high confidence in the selection of seed sets. In practice, developing a reasonable number of well-classified seed pages for an arbitrary topic can be considerably expensive. Moreover, the growth process from the original seed pages is not necessarily guaranteed to be productive, potentially requiring the iterative application of the web community discovery algorithm [9]. Therefore, in the third setting, in addition to having some seed pages, we assume that the user provides a set of representative keywords or terms that typify the desired community. We assign to each page a score that represents the value of its being part of the community. This is accomplished by using the given keywords to compute the relevancy score of each page based upon a lexical analysis of the page's content. These relevancy scores, possibly coupled with additional node scores derived from node importance and influence (say as in [12]), and weighted linkage-based relations among pages, can be then used to discover the community structure from the web graph.

The framework we choose for capturing web community discovery is the Random Field Ising Model (RFIM), widely used in statistical physics for the study of ferromagnetic materials. While it is convenient that the RFIM naturally encompasses both of our settings, it has other advantages as well. In particular, since one can find the solution for the given RFIM within the max-flow/min-cut framework (as demonstrated in [22]), we have a polynomial time algorithm that we can apply. Moreover, by subsequently applying a parametric flow approach to the model we can easily expand and contract the original extracted community. This notion is useful, for there is no “right” definition of a community and one sometimes has a practical target size to seek. We define a simple intuitive ranking scheme from the flow values that are produced during the application of the max-flow/min-cut algorithm. We conduct several experiments (both subjective and non-subjective) to validate the feasibility of our approach. Our experiments support the contention that appropriate link and node weighting and integration of various algorithmic approaches can greatly improve the quality in community extraction. This significantly improved quality is obtained without substantial increase in algorithmic execution cost and can be applied in a large and dynamically changing environment. In fact, various ideas presented in this paper have been integrated successfully into Brandimensions' web community system.

The need for efficiency in dealing with large scale graphs necessitates care in utilizing page content information. In particular, we do not want to consider the web graph as a complete graph with edge weights determined by some content similarity measure between all pairs of pages. Instead, we use co-similarity to seed pages as a form of implicit link, and then (following the rationale for co-citation) create additional links between two sites p and q when both p and q have sufficient similarity with the given seed pages. We also use site-seed similarity to adjust edge weights of existing links.

Our main contributions in this paper are listed as follows.

- We propose a RFIM-based framework for the extraction of viral communities by exploiting both the page content and the linkage-based relations of pages. This framework utilizes network flow algorithms and has been efficiently employed in a commercial application. The framework has additional practical aspects; in particular, it allows us to easily expand or contract the extracted community in a meaningful way.

- We propose a natural ranking scheme, *FlowRank*, for sites in the extracted community by using the flow values derived from the extraction process. Hence the ranking scheme is obtained with negligible additional computation cost.

- Using a concept of seed invariance, we show that our method is reasonably insensitive to the particular choice of seed nodes.

The rest of the paper is organized as follows. In Section 2, we give an overview of previous work related to the mining of communities from network graphs. In Section 3, we describe the RFIM-based models that we use for our blog community discovery algorithms. In Section 4, we present our inflation/deflation algorithms. In Section 5, we develop a novel way of ranking community members as a pleasant side-effect of the community extraction process. In Section 6, we present various experiments that we conducted to validate our approach. In Section 7, we present our conclusions including some possible extensions of our work.

2. RELATED WORK

Considerable effort has been expended upon deducing nascent and established communities of users from the web. Most work on the extraction of communities from a web graph is based on a graph partitioning approach. In this context, the goal is to examine each node and to perform a binary classification such that the cohesiveness among nodes in the included set is high with respect to the community being sought and low in the excluded set.

Flake *et al.* [9] use link analysis to construct a graph and then extract the community from the graph by solving the maximum flow problem. Clauset [7] identifies an approach using a greedy algorithm that infers local community structure from a known portion of a (possibly-larger) graph, as is appropriate for a crawler discovering links. Andersen and Lang [1] explore the problem of incubating seed sets into communities through random walks. Their approach uses a method of finding graph cuts by examining the sets determined by the random walk distribution at each step of the expansion process. The cuts are then improved by applying a maximum flow calculation to reduce the community cut size without materially impacting conductance. In our work, we treat the community discovery problem as one involving both connectivity analysis and content analysis, and build upon the success of using maximum flow to extract the final community.

Lin *et al.* [18] introduce the concept of blog communities as distinct geometries with the characteristic that bloggers are both producers and consumers of content. In this capacity, there is an important mutual awareness property that emerges in blog communities but is not present in typical web communities. This arises both from the bi-directional nature of the knowledge within blog communities and from differences in the semantic nature of blog hyperlinks. This is significant, because it emphasizes the sparseness of explicit hyperlinks in blogs, and hence implicitly boosts the importance of using other traits to define a complete community.

Hierarchical clustering has also been used as a mechanism for discovering community structure. Such methods rank and remove edges according to some measure of importance. Newman and Girvan [19] demonstrated a community extraction approach based upon centrality measures to define community boundaries.

Alongside this body of work in the web mining community, the theoretical computer science community has considered the metric labeling problem which seeks to find a classification that optimizes a combinatorial function consisting of assignment costs based on the individual choice of label for each object and separation costs between pairs [15]. It is known that the Random Field Ising Model presented in our paper is equivalent to the binary metric labeling problem. While the binary case can be solved in polynomial time, the problem becomes NP-hard when there are three or more labels.

There is a long history within the computer vision community of using the RFIM/metric labeling approach for the image segmentation problem (see [4] for a survey of various methods).

Part of our work relates to the ranking of pages within the extracted community. We focus on an approach whose results are materially different from classical link-based ranking algorithms such as PageRank [5] and HITS [13]. Whereas these Markov Chain-based algorithms rank web pages according to their link popularity (yielding hubs and authorities with HITS and a link-related conferred influence with PageRank), our approach uses the network flow model to define rank as a function of net residual flow through a node. A highly-ranked node (with high net residual flow) within our framework is not necessarily highly ranked by classical link-based ranking algorithms and *vice versa*. Motivated by [24], our flow-based model allows us to move beyond simple link-based ranking. We believe that our proposed approach is an inexpensive and natural way of merging lexical characteristics of web pages (heavily focused on blog and forum pages) with hyperlink information to produce a community-dependent ranking. As pointed out in [17], in the blogosphere, communities emerge because of the sustained action of contributors to blogs, not because of the informed or random navigation of readers.

3. THE BASIC MODEL AND SCE ALGORITHMS

We start this section with the description of the general Random Field Ising Model (RFIM) in the context of web community discovery. Using seed nodes, we will then proceed to use the RFIM to develop our SCE (*seeded community extraction*) algorithms.

3.1 Random Field Ising Model (RFIM)

Let $G = (V, E)$ be the graph representation of a subset of pages from which we want to extract our community structure. The weight function w_{ij} for undirected edge e_{ij} represents the similarity between pages i and j. Let h_i be the function that determines the likelihood of i's being a community member, and $\tilde{h}_i$ be the function that determines the opposite. The *web community discovery problem* is equivalent to finding a set $\delta = \{\delta_i \mid i \in V, \delta_i \in \{1, -1\}\}$ such that

$$H = -\frac{1}{2} \sum_{(i,j)\in E} w_{ij}\delta_i\delta_j - \sum_{\delta_i=1} h_i - \sum_{\delta_i=-1} \tilde{h}_i \qquad (1)$$

is minimized. Note that δ induces a binary node partition of graph G, yielding the desired community and non-community split. Let $X = \{i \in V \mid \delta_i = 1\}$ represent the derived community and let $\tilde{X} = \{i \in V \mid \delta_i = -1\}$ represent the derived non-community. Using (say) the Preflow-Push algorithm [11], the above minimization can be efficiently solved within the max-flow/min-cut framework.

3.2 Using Seed Nodes

In a generic "small" application of RFIM, one might expect that every node i has an associated likelihood h_i and that all edge weights w_{ij} are known or easily computable. However, we are considering applications involving very large graphs where reliable likelihood information about all nodes is not easily obtainable. Moreover, the size of such graphs makes it infeasible to consider complete graphs where all edge weights have been determined. However, our applications do permit a relatively small set of reliable seed nodes that we confidently know are in or not in the desired community. In this environment, we now proceed to show how we will utilize such seed information and how we will exploit semantic information about the sites that constitute the nodes of the graph.

Let μ be the set of *good seed pages* (members that should be included in the community) and let $\tilde{\mu}$ be the set of *bad seed pages* (members that should be excluded from the community). In the absence of keywords, the node weights (h_i and $\tilde{h}_i$) are only used to distinguish seed pages — both good and bad — from non-seed pages. We then choose to represent all possible features (both link-and content-based) that we consider for our web community discovery by the use of edge weights (w_{ij}). Our approach reinforces the similarity of pages based on semantic relations by exploiting edges from explicit hyperlinks and by selectively creating new edges. We believe that this is a natural extension of previous work that focuses on the information gleaned from links to represent similarity between pages. We let **SCE(weighted)** or more simply **SCE(W)** denote this algorithm which constructs weighted edges based on page semantics relative to our good seed pages. For the purpose of evaluating the benefit of this semantic information, we consider a base algorithm **SCE** that applies RFIM to the unweighted edge case (i.e. $w_{i,j} = 1, \forall(i,j) \in E$). In what follows, we first describe how to construct the node weights for SCE and SCE(W). We then describe how additional implicit edges and edge weights are constructed for SCE(W).

3.2.1 Node Weights for SCE and SCE(W)

We construct node weight values as follows:

$$h_i = \begin{cases} \text{K} & \text{if } i \in \mu \\ 0 & \text{Otherwise} \end{cases} \qquad \tilde{h}_i = \begin{cases} \text{K} & \text{if } i \in \tilde{\mu} \\ 0 & \text{Otherwise} \end{cases}$$

where $K = max_{i\in V} \sum_{\{j|w_{ij}\neq 0\}} w_{ij}$. It is proven in [4] that such choice for K guarantees that all good seed pages are included in the extracted web community while all bad seed pages are excluded from it.

3.2.2 New Edges and Edge Weights for SCE(W)

Edge weights are used to reflect the similarity between two pages in the web graph, taking into account both linkage and content relations between the pair. Semantic information is combined with link information. In so doing, page content is used to reinforce the relation of two pages if there is an explicit link between them, while an implicit link (created from semantic relations) is generated if there is no pre-existing edge. Note that this requires a complete graph construction. In the next subsubsection, we propose an approach to avoid such a complete graph construction. To compute the content-based similarity between two pages, we first parse each page with respect to the extracted features to produce a canonical vector representation of the page. The features that we consider are page content, title, metadata (description and keywords), and anchor text, all of which have been used in other web mining applications (e.g. [8]). To construct the term frequency-inverse document frequency (TF-IDF) vector representation of each page's features, we perform the following pre-processing: (1) For page content and title, we first eliminate stop words and then further conflate remaining words using the standard Porter Stemmer [21]. We reduce the term space dimension even further by using document frequency thresholding (DF) [26] to de-emphasize the impact of rare terms unlikely to influence global performance. (2) For metadata and anchor text, we perform similar pre-processing operations except that these features bypass stemming. We also massage the link list by removing all nepotistic links. We employ the extended Jaccard coefficient (Tanimoto similarity measure) for computing the similarity between various string data objects, as this metric has been shown to produce superior results for various clustering approaches

Algorithm 1 Similarity Approximation Algorithm

```
Compute $\sigma_g(\mu^g, p)$ for every page $p \in V$ using centroids.
for all $p \in V$ do
  if $S(\mu^g, p^g) \geq \delta$ then
    for all $q \in V$, $q \neq p$ do
      if $S(\mu^g, q^g) \geq \delta$ then
        $\sigma_g(p, q) = S(\mu^g, p^g) \cdot S(\mu^g, q^g)$
      end if
    end for
  end if
end for
```

[23]. The extended Jaccard coefficient for pages p_1 and p_2 with respect to a feature g (e.g. meta description) is defined as

$$\sigma_g(p_1, p_2) = \frac{p_1^g \cdot p_2^g}{|p_1^g|^2 + |p_2^g|^2 - p_1^g \cdot p_2^g}$$

where p_i^g is the TF-IDF vector representation of feature g on page i. Using this measure, we can compute the similarity between each pair of pages i with respect to different features. Finally, we combine all similarity values associated with each page as a weighted linear sum to produce a single similarity value, w_{ij}, between each pair of pages p_i and p_j:

$$w_{ij} = \sum_{g_k \in \Omega} \sigma_{g_k}(p_i, p_j) \cdot \phi_k + \omega_l$$

where Ω refers to the features we consider, ϕ_k is a suitable weight for each $g_k \in \Omega$, and w_l is a weight used to reinforce the final similarity value if there is a hyperlink between p_i and p_j or 0 otherwise. While it is possible to use more sophisticated techniques for combining similarity measures, we leave this as a topic for future research.

3.2.3 *Constructing a Reasonably Sparse Graph*

The edge construction approach just described requires the content similarity computation of every pair of nodes in the dataset, and consequently the possible construction of a complete graph (as it is possible to have an implicit link for every pair). This is not feasible nor necessarily desirable if we have considerable confidence in the quality of our seed nodes. Therefore, in order to exploit the use of seed nodes and to dramatically improve algorithmic efficiency, our approach will be to only compute direct similarity between seeds and other pages. We will then construct new edges between pages p and q if and only if both p and q are "similar enough" to good seeds.

Our intuition is similar to that of the use of co-citation relations. Namely, we can view the lexical similarity of a page p to the good seed nodes as a probability that p is in the desired community. If we think of pages as being constructed independently (which, in general, is not the case) then the probability that both p and q are in the desired community becomes the product of these probabilities. On the other hand, some amount of common usage of terminology will provide a small measure of similarity even when none may exist. Hence our approach is to infer a semantic relation (and hence an implicit link) between p and q if and only if both pages exceed a minimal amount of lexical similarity to good seed pages.

More precisely, let p^g be the TF-IDF vector representation of page p with respect to content feature g. Let $\mu^g = \{\mu_1^g, \ldots, \mu_{|\mu|}^g\}$ denote the set of TF-IDF vector representations of seeds in μ with respect to content feature g. Then, let $S(\mu^g, p^g)$ be the similarity measure between the seeds and page p. There are various options for $S(\cdot, \cdot)$. For instance, $S(\mu^g, p^g)$ can be defined as the average distance between each element in μ^g and p, or as the minimum of all distances between each element in μ^g and p^g. We begin by first constructing implicit links, weighted by $S(\mu^g, p^g)$, between μ^g and page p. The similarity $\sigma_s(p, q)$ for pair p and q is then constructed assuming $\sigma_g(p, q) \propto S(\mu^g, p^g) \cdot S(\mu^g, q^g)$ if both $S(\mu^g, p^g)$ and $S(\mu^g, q^g)$ are sufficiently close to seed pages (i.e. both $S(\mu^g, p^g)$ and $S(\mu^g, q^g)$ are lower bounded by a constant δ). The algorithm is summarized in Algorithm 1.

3.2.4 *Speeding Up the Similarity Computations*

Using our approach, we compute the similarity between every pair of elements that is semantically related (with respect to the seeds), and then apply these similarity values to construct the edge weights. This will still require $O(|V||\mu|)$ similarity computations. However, to further reduce the complexity of the algorithm, we can compute $S(\mu^g, p^g)$ with respect to the *centroid* of the given good seeds. By constructing the centroid of seed pages, μ_c^g, as the average of all good seeds in μ^g, we can simply take $\sigma_g(\mu_c^g, p)$ as $S(\mu^g, p^g)$ for every page $p \in V$.

3.3 Extending the Method to Exploit Keywords

The second problem setting that we consider is where the user knows (or can divine) a set of representative keywords or terms for the desired web community to augment the seed pages. Note that this problem formulation might arise in a number of ways. For example, the user might not be fully confident about the quality of the provided seeds, or it might simply be too expensive to provide many. Our hypothesis is that we can improve the community extraction process if the user can provide some keywords that are representative of community members. For instance, the user interested in the Mustang car community might provide keywords such as *GT500* and *Mustang* to help disambiguate commonly-occurring words in the intended community. In practice, it would be almost impossible for the user to develop a complete set of representative terms for the community.

For this particular setting, we construct node weights (h_i) and edge weights (w_{ij}) as follows. For each potential candidate (excluding seeds) of the community, we assign a "relevancy" score with respect to the set of keywords provided by the user. We then associate the content-based features (page content, title, and metadata) of each page with its node weight, and we associate the link-based features (including anchor text) between two pages with their edge weight. As before, the semantic co-citation between pages is used to reinforce or possibly create edge weights. The resulting algorithm is called **SCE(weighted + keywords)** or simply **SCE(W+K)**. In the following, we describe how to construct appropriate node and edge weights.

3.3.1 *Node Weights*

Node weight construction for seed pages the same as for SCE and SCE(W). For the rest of nodes, we assign the likelihood of being part of the targeted community as follows. The page parsing and pre-processing steps are similar to those described for SCE(W). We represent the given keywords associated with the putative community to be mined as a vector normalized with the inverse document frequency of the terms (with respect to the page feature) obtained from the given corpora. Let q be such a vector representation. Given q, we compute the relevancy score for feature g of page j as

$$R(q, p_j^g) = \sum_{i=1}^{t} (q_i \cdot u_{ij}) \qquad (2)$$

where $u_{ij} = \frac{1+ln(1+freq_{ij})}{ndl} \times IDF_i$, $freq_{ij}$=the frequency of the term i within the page p_j's feature g, IDF_i=an estimate of the inverse document frequency of the term i in the corpora (with respect to feature g), and $ndl = (1 - \sigma) + \sigma \cdot \frac{dl(p_g)}{avgdl}$ with $dl(p_g)$ referring to the document length of page p_g, $avgdl$ referring to the average document length in the dataset, and $0<\sigma<1$ being a constant. Finally, we combine relevancy scores for all features as a weighted linear sum to produce a single relevancy score, $R(p_i)$ for each page p_i. We translate the relevancy scores of pages into h_j values using the following relations:

$$h_j = R(p_j) \quad \tilde{h}_j = \begin{cases} \alpha & \text{if } R(p_j) = 0 \\ 0 & \text{Otherwise} \end{cases}$$

where α is a weight, normally chosen to be small, to assign a constant value likelihood of not being in the community to those pages that do not contain the given seed keywords. The proposed node weight construction is almost identical to that of the content-based relevancy scores used by the standard algorithmic web search engines to retrieve web pages relevant to a given query.

3.3.2 Edge Weights

SCE(W+K) and SCE(W) have the same edge set. However, we boost the edge weights of explicit links between pages depending on the presence of the given keywords in the corresponding anchor text. Let s^l_{ij} be

$$s^l_{ij} = \begin{cases} 1 + \epsilon \cdot R(a_{i \to j}, q) & \text{if } i \to j, \neg(j \to i) \\ 1 + \epsilon \cdot (R(a_{i \to j}, q) + R(a_{j \to i}, q)) & \text{if } i \to j, j \to i \\ 0 & \text{Otherwise} \end{cases}$$

where $i \to j$ denotes a link from i to j, and $R(a_{j \to i}, q))$ is the relevancy score, computed using Eq. 2, for the anchor text associated with the link $j \to i$. Note that a similar approach was taken by Chakrabarti *et al.* in [6]. Finally, we combine all similarity values (both link- and content-based) associated with each page as a weighted linear sum to produce a single similarity value, w_{ij}, between each pair of pages p_i and p_j:

$$w_{ij} = \sum_{g_k \in \Omega} \sigma_{g_k}(p_i, p_j) \cdot \phi_k + \omega_l \cdot s^l_{ij}$$

Once again, we can accelerate the construction of edge weights using the centroid heuristic proposed in Section 3.3.

4. CHANGING THE SIZE OF EXTRACTED COMMUNITY

While the previously-defined SCE algorithms produce a community structure of the given web graph, it is possible that the resultant community does not match what the user originally expected since, in practice, the notion of optimal community is somewhat subjective. For instance, for the discovery of the web community associated with the Mustang car, one may or may not accept a page discussing stores that sell accessories for Ford cars. Therefore, in this section, rather than trying to fully automate the process of web community discovery, we present the tools for *inflating* or *deflating* the already-discovered community structure. By applying our tools, the user can choose a web community that is optimal according to his personal view of what the community actually represents. In fact, multiple versions of the community are sometimes appropriate, depending upon the desired size and level of cohesion sought.

Algorithm 2 Inflation Algorithm

```
INPUT: $m$, $(X, \tilde{X})$
OUTPUT: $\Omega$
$\lambda = 0, \Omega = \emptyset, \forall\, v \in V, d(v) = 0$
while $(m > \lambda)$ do
    for all (v,t) ∈ E do
        $f(v, \lambda + 1) = min\{c_{(v,t)} - \lambda, f(v, \lambda)\}$
    end for
    for all (s,v) ∈ E with $d(v) < n$ do
        $f(v, \lambda + 1) = max\{c_{(v,t)} + \lambda, f(s, \lambda)\}$
    end for
    Run Preflow(f,d)
    for all $v \in V$ do
        $d(v) = min\{d_f(v, s) + n, d_f(v, t)\}$
    end for
    if $(X_\lambda, \tilde{X})$ is inflated then
        $\Omega = \Omega \cup (X_\lambda, \tilde{X})$
    end if
    $\lambda$++
end while
Return $\Omega$
```

Given the binary partition $(X, \tilde{X})$ of G, let $(Y, \tilde{Y})$ be the inflation of $(X, \tilde{X})$ if $X \subset Y$ and $\tilde{Y} \subset \tilde{X}$ such that $Y \sqcup \tilde{Y} = G$. Similarly, let $(Z, \tilde{Z})$ be the deflation of $(X, \tilde{X})$ if $Z \subset X$ and $\tilde{X} \subset \tilde{Z}$ such that $Z \sqcup \tilde{Z} = G$. In Gallo *et al.* [10], a way of performing the inflation or the deflation of the given $(X, \tilde{X})$ is provided using the *parametric network* framework. In a parametric network of the st-graph, the arc capacities are functions of a real-valued parameter λ. We denote the edge weight function by w_λ and make the following assumptions about edge e_{ij}:

- $w_\lambda(e_{sv})$ is a non-decreasing function $\lambda\ \forall v \neq t$.
- $w_\lambda(e_{vt})$ is a non-increasing function $\lambda\ \forall v \neq s$.
- $w_\lambda(e_{vw})$ is constant for all $v \neq s$, $w \neq t$

In [10], it is proven that there exists a parametric flow algorithm that can efficiently produce partitions, $(X_i, \tilde{X}_i)$, $i > 0$, where $X_i \subset X_{i+1}$ and $\tilde{X}_{i+1} \subset \tilde{X}_i$ holds as the value of λ increases. Therefore, we have $\lim_{\alpha \to \infty} X_\alpha = G$ and $\lim_{\alpha \to \infty} \tilde{X}_\alpha = \emptyset$. This algorithm is particularly attractive as it allows computing a chain of min-cuts at the cost of only a constant factor (relative to that of the Preflow-Push algorithm) in its worst-case time bound. By adapting this parametric flow idea for our particular application domain, we develop a parametric flow algorithm that recursively computes a chain of min cuts, $(X_1, \tilde{X_1})$, $(X_2, \tilde{X_2})$, $\cdots$, $(X_l, \tilde{X_l})$, such that the constraint $X_1 \subset X_2 \subset \cdots \subset X_l$ is held when the value of λ is incremented by 1 each time. We present the outline for our Inflation algorithm in Algorithm 2. In our algorithm description, $f(v, \lambda)$ refers to the flow value at iteration λ, $d_f(v, w)$ refers to the distance from node v to node w in the current residual graph with respect to flow f, and $Preflow(f, d)$ refers to the Preflow function used for the classical Preflow-Push algorithm [11]. The corresponding deflation process easily follows. The parametric network framework is particularly attractive due to its computational efficiency, as it is able to incrementally produce a chain of either inflated or deflated communities.

5. RANKING OF COMMUNITY MEMBERS

Even after we have extracted a community structure from the given web graph, it is quite expensive to evaluate the true quality of the community members so identified. We wish to understand, for example, how visible or influential a page is in the context of our extracted community. Furthermore, when we want to perform a retrieval task over the community (e.g. retrieve all community members that are relevant to a query), it is necessary to come up

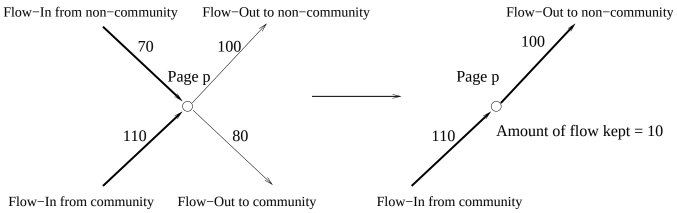

Figure 1: An illustration of our FlowRank approach

with a ranking scheme to assess the importance of each retrieved page and so order the query results. In this section, we present a simple yet intuitive ranking algorithm which arises as an intrinsic component of our community extraction process. We call our ranking scheme *FlowRank*.

Since our SCE algorithms are based on the maximum flow-minimum cut framework, the flow f_{ij} produced between two pages i and j can be viewed as the exchange of authority between the pages. Moreover, if $f_{ij} > f_{pq}$, then this implies that i and j are more actively exchanging authority than are p and q. Tomlin follows a similar line of thinking, and defines the sum of flow values into (or out of) a page as the ranking value corresponding to a page's traffic [24]. We extend this concept and apply it to rank members in our extracted community[4]. Therefore, the rank of a page that is part of the community can be interpreted as the *reputation* of the page within the community. The reputation is expressed as a combination of flows from/to other community members and flows from/to non-community members.

Note that in any st-graph, the condition,

$$\sum_{i \in G} f_{ip} = \sum_{j \in G} f_{pj}$$

holds for every node $p \neq s, t$. Since the flow values come from two different sources, the given condition can be rewritten in terms of flows from the community and flows from the non-community.

$$\sum_{i \in X} f_{ip} + \sum_{i \in \tilde{X}} f_{ip} = \sum_{j \in X} f_{pj} + \sum_{j \in \tilde{X}} f_{pj}$$

Intuitively, the flow into page p from other community members can be seen as an endorsement of authority from these other community members. Therefore, if page p has higher flow values into it from other community members compared with page q, then the rank of page p (with respect to the given community) should be boosted more than q. However, page p also emits flows toward other non-community members. Accordingly, if page p has higher flow values from it into other non-community members compared with page q, then the rank of page p (with respect to the given community) should be penalized more than q. In summary, the amount of flows from other community members that are kept by nodes should be taken as the ranking value for page p (See Figure 1). More formally, let $FR(p)$ denote the rank of page with respect to the given community $G = (X, \tilde{X})$, which will be defined as

$$FR(p) = \sum_{i \in X} f_{ip} - \sum_{j \in \tilde{X}} f_{pj} = \sum_{j \in X} f_{pj} - \sum_{i \in \tilde{X}} f_{ip}$$

Intended Community	Number of Good Seeds	Number of Bad Seeds	Keywords Used
Camry	111	22	toyota, camry, car automobile, auto, hybrid
Mustang	409	19	mustang, ford, GT500 car, automobile, auto
Ipod	124	23	ipod, shuffle, nano, apple, itunes
Playstation	291	14	playstation, psp, game playstation2, psx,ps2
Xbox	139	15	xbox, xbox360 microsoft, game

Table 1: Summary of intended communities and their respective terms and seeds

	Average Degree
Original Graph	2.1147
Camry	3.6679
Playstation	5.8345
Mustang	12.5575
Xbox	14.9933
Ipod	11.3174

Table 2: Average degrees before and after similarity induced links

Since FlowRank emerges as a consequence of our extraction process, it affords us with a natural way to incorporate additional features such as content into the ranking scheme. Moreover, FlowRank can be used for both query-dependent (computed online) and query-independent (computed off-line) rankings with respect to a community in the following sense: Since our SCE(W+K) benefits from a having a set of representative keywords for the community (which can be interpreted as a query string provided by the user), FlowRank can easily be adapted to be the query-dependent ranking component of a community-targeted search engine. When FlowRank is used in the context of our first scenario with seed pages, on the other hand, it can be used as the query-independent ranking component. This is especially attractive as the additional cost associated with the computation of FlowRank is negligible once the web community extraction has been performed.

6. EXPERIMENTS

In this section, we describe the experiments that we conducted to test our approach. The first set of experiments evaluates the quality of the community extraction. The next experiment analyzes the performance of FlowRank for ranking members in the community. We then consider a particular stability issue, that of "seed invariance". The last experiment looks at the behavior of our inflation/deflation algorithm.

6.1 Description of Dataset

To run our experiments, we took a random subset [5] of approximately 2.84 million blog and forum entries from a Brandimensions database. Every day, around 1 million new entries from different blog and forum sites covering a vast range of different industry sectors including automotive, entertainment, gaming, consumer electronics, and pharmaceuticals, were collected and stored in the database. We extracted 5 different communities, namely `camry`, `ipod`, `mustang`, `playstation` and `xbox`, from the

[4] Our approach can easily be extended to rank non-community members as well. Even with the ability to inflate or deflate a community, it is sometimes useful to be able to view the top candidates that did not make the cut.

[5] Our dataset was constructed during January, 2007. Experimental data can be found at `http://www.affsys.com/experiments/HT2008`.

	SCE			SCE(W)			SCE(W+K)		
Dataset	comm. size	precision	time	comm. size	precision	time	comm. size	precision	time
Camry	64936	15%	9.26 sec	1371	57%	15.51 sec	712	60%	15.00 sec
Ipod	73681	17%	9.15 sec	3612	51%	51.68 sec	2052	77%	49.44 sec
Mustang	65388	13%	9.07 sec	3695	51%	57.21 sec	1379	83%	55.65 sec
PStation	75081	26%	8.54 sec	2255	64%	24.29 sec	1315	80%	22.23 sec
Xbox	93965	32%	9.52 sec	5056	50%	71.84 sec	1265	84%	69.26 sec
Average	74610.2	**20.6%**	9.108 sec	3197.8	**54.6%**	44.106 sec	1344.6	**76.8%**	42.316 sec

Table 3: Summary of Community Extraction

Mustang	
http://www.mustangforums.com/m_1112404/tm.htm	-Shelby GT500 Allocation
http://www.motorsportsblog.com/.../mustang_muscle.php	-Motorsports Blog: MUSTANG MUSCLE FOR RENT
http://www.autoblog.nl/.../ford-mustang-concept-door-giugiaro	-Mustang Concept Door Giugiaro
http://lovethemustang.blogspot.com/......./ picture-of-07-mustang-shelby-cobra.html	-I love the Mustang GT: Picture of the 07 Mustang Shelby Cobra GT500
http://forums.stangnet.com/showthread.php?t=450785	-DOHC conversion on a fox body..- Mustang Forums at StangNet
http://forums.stangnet.com/showthread.php?t=625052	-2007 CorvetteZ06 Vs 2007 GT500- Mustang Forums at StangNet
http://www.fordforums.com/showthread.php?t=103002	-US:Wild Mustangs: Tuner XMP builds Crazy Horse II; Shelby's 'other' GT500

Ipod	
http://forums.ipodlounge.com/showthread.php?p=1033468#post1033468	-How to coil my cord? -iPod Forums at iLounge
http://forums.ipodlounge.com/showthread.php?t=24841	-top 15 most played songs on your ipod- iPod Forums at iLounge
http://forums.techguy.org/.../425050-ipod-mini-not-charging-right.html	-Ipod Mini not charging Right -Tech Support Guy Forums
http://forums.ipodlounge.com/showthread.php?t=161707	-Hooking Up A 30G Video → 2005 Nissan Sentra-iPod Forums at iLounge
http://ipodnewsblog.blogspot..../ new-dodge-caliber-features-aux-jack.html	-New Dodge Caliber features aux jax and iPod holder
http://new4uu.blogspot.com/2006/08/best-freeware-ipod-utilities.html	-Best freeware ipod utilities
http://freewaremac.blogspot.com/2006/03/ipod-hi-fi-review.html	-FreewareMac: iPod Hi-Fi review

Table 4: Sample community members for `Mustang` and `Ipod` (using SCE(W))

given dataset. The task of extracting these 5 communities was challenging due to the random composition of the data and the mixture of similar and diverse communities in the Brandimensions database.

6.2 A Subjective Evaluation

In our first experiment, we ran the community discovery algorithms described in Section 3. Table 1 summarizes the inputs that we used for each community discovery task. Seeds were used for all SCE algorithms, while keywords were used only for the SCE(W+K) algorithm. In Table 2, we report on the average degree (ignoring direction) of the graphs constructed in SCE(W) and SCE(W+K). Table 2 shows that the edge density was considerably increased by incorporating egdes induced by lexical similarity.

We implemented and ran our SCE algorithms in C++ on a Linux-Based machine with a 2.4 GHZ processor and 8 GB RAM. We used HIPR[6] to find max-flow/min-cut solutions. Some important parameter values that we employed for our experiments were $\alpha = 0.8$, $\epsilon = 0.15$ and $\delta = (2.5) \cdot\ A\_sim$ where A_sim refers to the average $S(\mu_g, p^g)$ value for each dataset [7]. The evaluation of extracted communities was done by three individuals from Brandimensions, two of whom were professional categorizers who had extensive experience evaluating this type of data. The third person had no particular categorization skills, but was given the same instructions regarding how to evaluate the result sets. We took the average of the values reported by the categorizers to compute the numbers reported herein. One hundred randomly-chosen pages of each extracted community were shuffled and then evaluated by each categorizer. Without any prior knowledge about what algorithm was used to produce the corresponding result, each categorizer was asked to carefully classify each page as "relevant" if, in their judgment, the page should be treated as a member of the corresponding viral community, or "non-relevant" otherwise. In Table 3, we summarize results of each extraction task. In Table 3, *comm. size* refers to the size of the extracted community, *quality* (i.e. precision) refers to the portion of members out of 100 samples that were classified as relevant, and *time* refers to the execution time for the extraction. The first observation that we can make is that our SCE(W) and SCE(W+K) algorithms employing semantic analysis of content outperform the more purely link-based community extraction method SCE. This supports our hypothesis that the semantics between members of a community are important to consider when capturing the essence of the community. We argue that the link from a blog/forum entry to another blog/forum entry cannot always be translated into an endorsement of authority from the source to the destination, as is the case with classical web pages.

[6] http://www.avglab.com/andrew/soft/hipr.tar

[7] Our choice of parameters is rather arbitrary and subjective. For example, in setting α, we note $0 \leq R(p_j) \leq 1$ and it seems that a page not containing any of the keywords is very likely not to be in the community.

Community: Camry	
FlowRank	PageRank
RC car, but its a Toyota Camry... Focaljet... http://forums.focaljet.com/showthread.php?t=428466	2007 Toyota Camry Official Configurations, Specs, and Photos http://autoblog.com/.../2007-toyota-camry-...-photos
Toyota Camry-Our Latest Road Test Articles http://eamon.blogfa.com/post-32.aspx	SUV & Truck Forum at Truck Trend Magazine http://forums.trucktrend.com
Toyota Camry Blog Archive US: GM interiors get stylish http://toyotacamry.blogsautos.../us-gm-interiors-get-stylish	KickingTires: Suburband Dad: 2007 Toyota Camry http://blogs.cars.com/.../suburban_dad_20.html
Official: New Camry and Hybrid Camry-Ford Australia Forum http://www.fordaustraliaforums/.../showthread.php?t=15490	Toyota Kentucky Plant about to build five-millionth Camry http://feeds.autoblog.com/.../3̃/22751781
86 Toyota Camry-broken timing belt http://car-forums.com/talk/showthread.php?t=5733	Auto Lah-Auto Industry News: New Toyota Camry and Avanza http://autolah.../new-toyota-camry-and-avanza.htm

Community: Playstation	
FlowRank	PageRank
Free Sony Playstation 3 . . . Playstation 3 Console Pre-Order Update http://free-playstation-....blogspot.com/...-pre-order-update.html	GamersVue http://gamersvue.blogspot.com
Why the Nintendo Wii's price is not excessive http://www.ryansgoblog.com.../why-the-nintendo-wiis-price-is-not-excessive	CamersVue Re/PreVue http://gamervuevues.blogspot.com
Free Sony Playstation 3 . . . List of Sony Playstation 3 Games http://free-playstation.../list-of-sony-playstation-3-games.html	GamersVue-Playstation http://gamersvueplaystation.blogspot.com
GamerC: Only bullies play the Playstation 2 http://gamerc.blogspot.com/2006/08/only-bullies-play-playstation-2.html	GamersVue Q & A http://gamersvueqanda.blogspot.com
The Console Wars: Square Enix Xbox 360 Update http://theconsolewars.blogspot.com/2005/07/square-enix-xbox-360-update.html	Generation Star Wars http://johnhood.blogspot.com

Table 5: Top 5 ranked members by FlowRank and PageRank for `Camry` and `PlayStation` (using SCE(W+K))

For instance, we anecdotally observed that many bloggers link to their parental blog site. Furthermore, the fact that a blogger has left a comment in another blog frequently does not imply that they are in the same viral community. To provide a more concrete view of our extracted communities, we present some samples (chosen at random) of community members (excluding seed pages) that are produced for the SCE(W) algorithm in Table 4. These samples again validate the quality of our extracted communities.

Note that the SCE(W) algorithm judges the inclusion or exclusion of a page in the community based on its lexical similarity to other members of the community. We found that many times this was not sufficient as it could lead to the inclusion of some pages (although not usually ones highly ranked by FlowRank) that are not strongly related to the community topic. This is especially true as our intended experimental communities are very targeted ones. For instance, we have found that for both the `Mustang` and `Camry` communities, the SCE(W) algorithm returned a considerable number of pages related to the general automotive industry but not specifically related to `Mustang` or `Camry`. We observe a similar phenomenon with `Ipod`, `Xbox` and `PlayStation`. For `Ipod`, various pages related to Ipod but not strongly related to the Ipod community were retrieved as community members, while for both `Xbox` and `PlayStation`, numerous game-related sites not strongly on `Xbox` or `PlayStation` were retrieved as community members. Our algorithm sometimes failed to detect blog spamming. On the other hand, the SCE(W+K) algorithm tends to minimize such effects as it performs a more refined content analysis for each page to determine its eligibility. When SCE(W+K) does misclassify a site, we attribute this failure to the particular choice of keywords. For instance, several pages related to Ford but not directly to Ford Mustang were retrieved as part of `Mustang`, due to the inclusion of the keyword Ford as input. Another interesting observation was that several communities overlapped. For instance, we observed that various discussions on `Xbox`, `Playstation` and `Revolution`[8] were found to be mixed together across the `Xbox` and `Playstation` communities. In particular, we found numerous blog entries from these communities comparing some of the three game consoles. Consequently, this can be considered a failure of any extraction algorithm since the strongest discussion topic within the corresponding blog/forum entry does not necessarily match the intended community[9].

6.3 Ranking Results

In this section, we study the results of our ranking produced by FlowRank. We compute the ranking with respect to each web community extraction approach. We also devise a simple ranking scheme based on PageRank and compute the ranking for each web community approach for the sake of comparison. Specifically, we computed the PageRank values for all nodes in our dataset. Using their respective PageRank values, we further order all members of each extracted community. For brevity, we report only on the results of the top 5 members from the `Camry` and `PlayStation` communities produced using SCE(W+K) in Table 5. PageRank-based rankings and FlowRank-based rankings are considerably different in nature. These top-ranked members again indicate the quality of the extracted web communities produced by SCE(W+K) since all pages returned by FlowRank are authoritative pages in their corresponding community. Furthermore, these examples suggest the superiority of our FlowRank algorithm for the particular application of ranking community members. For instance, for `PlayStation`, most of the top-ranked pages by PageRank are not strong authoritative community members (in fact, most of them are non-community members) while for `Camry`, one of the top-ranked pages by PageRank (`http://forums.trucktrend.com`) is not strongly related to `Camry`. This is due to the fact that PageRank tends to prefer pages at or close to the root of a site, likely because of the influence conferred upon such pages by the link structure of the web. On the other hand, FlowRank tends to consider semantic tightness of each page to the extracted community to produce its ranking, thereby resulting in a better ranking quality.

[8] A game console developed by Nintendo

[9] Certainly one can subjectively argue that these types of pages should be classified as community members as well, but this was not the choice made by our categorizers who were relatively strict in evaluating our extracted communities.

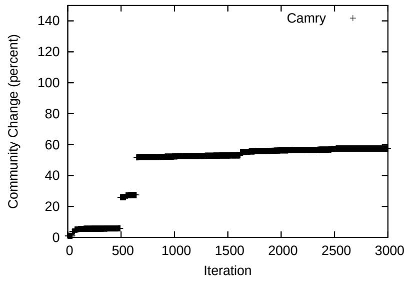

Figure 2: Change in Community Size (Camry)

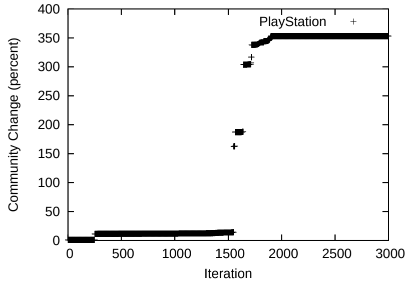

Figure 4: Change in Community Size (PlayStation)

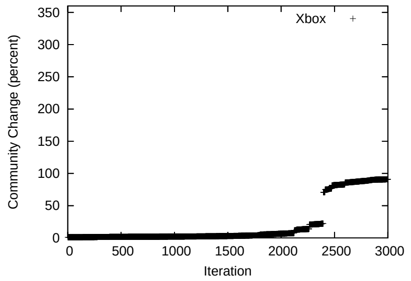

Figure 3: Change in Community Size (Xbox)

6.4 Inflation/Deflation Aspects

To understand the dynamics of our extracted communities, we ran the parametric flow algorithm over these communities. Due to space limitations, we present only the results for `Camry` (Figure 2), `Xbox` (Figure 3) and `PlayStation` (Figure 4). For each community, we varied the value of λ defined in Section 4, and we grew each community starting from a considerably small size (say only 10 community members) recording the community size change (with respect to the original community size) for each iteration. In all figures, the X-axis corresponds to the range of λ values, while the Y-axis corresponds to the change of community size with respect to the original community size. Observe that the rate of increase in community size slows down after a certain number of iterations, resulting in a stable structure. The near step-function response arises from a bottleneck in a community cluster boundary that the parametric flow algorithm abruptly overcomes. This boundary indicates a clear transition point across which few connections reach further out into the graph compared with those reaching in. These sharp transitions seem to reflect natural levels of community cohesion and we are currently investigating how better to understand (both experimentally and analytically) the nature of such transitions.

6.5 Seed Invariance

Intuitively, we would like to understand how sensitive our results are to the choice of the seed set and more particularly the set of good seeds. There are various ways to study this concept and in this section we briefly describe one experiment in this regard.

Let ALG be any of the SCE algorithms. We let C_{ALG} denote the community extracted by ALG when applied to some given target community with say g good seed nodes. For $0 < f \leq 1$, we let $C_{ALG}^{\rho(f)}$ denote the community extracted when we replace a random subset of $s = f \cdot g$ initial good seeds by a random subset of C_{ALG} of size s. Similarly we let $C_{ALG}^{\tau(f)}$ denote the extracted community when the initial good seed nodes are replaced by the s top-ranked (by FlowRank) members of C_{ALG}. In Table 6 we report on the case $f = 1$ where all initial good seeds are removed and for notational simplicity we use C^{ρ}_{ALG} and omit the f and use C^{ρ}_{ALG} and C^{τ}_{ALG}.

Using 10 random trials, we computed the Jaccard similarity measure between C_{ALG} and C^{ρ}_{ALG}, and between C_{ALG} and C^{τ}_{ALG}. One can think of this either as a test of seed invariance or as a non-subjective test of the performance of the algorithm. We observe that all of the algorithms appear to be quite seed invariant. Moreover, in all cases, the similarity measure improves when using the top-ranked extracted sites rather than random extracted sites. We view this as evidence that FlowRank is providing an informative ranking. However, the results for SCE seem at first counter-intuitive. Given the subjective evaluation for the quality of the extracted communities, one might expect $SCE(W + K)$ to be more seed invariant than $SCE(W)$ which in turn would be more invariant than SCE. To explain this seemingly counter-intuitive behavior, we note that an algorithm whose quality (i.e. precision) is only (as for CAMRY) 15% is still able to insert (an expected) $.15g$ good community members as replacements for the initial good seeds. If the other 85% of the misclassified nodes are not having a coherent impact on the results, then an algorithm like SCE can still be very seed invariant by consistently returning essentially the same (perhaps relatively low precision) community. Looking at the induced degree structure[10] for the SCE-extracted communities shows that the SCE communities are quite dense and hence just having a few seed nodes within these tightly linked induced subgraphs will result in the same extracted community. The test we have devised will return strong similarity values if either the algorithm is insensitive to a large amount of noise in the seed set or if the community extraction results were very good. With its minimal dependence on seeds and the dense structure of the resulting communities, SCE exhibits better "seed invariance". The different results for SCE(W) and SCE(W+K) are due to their reliance on seeds without having the same quality of results.

7. CONCLUSION

We have explored the benefit of page semantics in the discovery of viral communities from a given graph. Based on the RFIM model, we proposed two different problem settings and showed how the community mined from a web graph could be fine-tuned through the use of parametric flow. We also proposed a way of ranking the community members from the flows produced as an outcome of the community extraction process. Our preliminary experiments indicate that the quality of extracted blog communities using our more semantic approach is better than that obtained

[10] The average degree for the SCE-extracted communities varied from 25.8 to 33.9.

C^{ρ}_{ALG}						
Algorithm	Camry	Mustang	Ipod	Playstation	Xbox	Final
SCE	0.874532873	0.697045371	0.799871536	0.882362604	0.706890864	0.79214065
SCE(W)	0.366153072	0.482200014	0.497170059	0.577346068	0.669444444	0.518462731
SCE(W+K)	0.765443151	0.482469911	0.827127952	0.804461942	0.667749671	0.709450526
C^{τ}_{ALG}						
SCE	0.992546507	0.990974607	0.979552823	0.976225676	0.712733465	0.930406616
SCE(W)	0.992205438	0.636585429	0.993286855	0.984068612	0.802921811	0.881813629
SCE(W+K)	0.818263205	0.717425432	0.862932455	0.843996063	0.661406359	0.780804703

Table 6: Seed Invariance Results

through mainly pure link-based approaches. One very promising future direction follows the insightful temporal analysis work of Kleinberg [14]. We can apply time sequence analysis to blogs or postings within a blog to obtain further ranking information of sites within a discovered community. We are in the process of using such time sequence analysis to discover additional implicit links between sites as well as to reinforce the weights of existing links so as to improve the community extraction process and study its dynamic behavior (e.g. how fast communities evolve, and how often new influential sites emerge). We are also studying how FlowRank compares with the work of Kempe, Kleinberg and Tardos [12] for discovering influential sites within a community. We believe that community similarity tests as suggested in Section 6.5 can be used to help adaptively refine the selection of keywords and seed nodes. Finally, we are considering alternatives to our seed invariance measure so as to provide improved non-subjective evidence for the quality of an extraction algorithm and for evaluating the quality of community ranking functions.

8. REFERENCES

[1] R. Andersen and K. J. Lang. Communities from seed sets. In *WWW*, pages 223–232, 2006.

[2] R. Angelova and G. Weikum. Graph-based text classification: learn from your neighbors. In *SIGIR*, pages 485–492, 2006.

[3] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR*, pages 104–111, 1998.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[6] S. Chakrabarti, B. Dom, R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. M. Kleinberg. Mining the web's link structure. *IEEE Computer*, 32(8):60–67, 1999.

[7] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72:026132, 2005.

[8] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, 2003.

[9] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD*, pages 150–160, 2000.

[10] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, 1989.

[11] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. In *STOC '86*, pages 136–146. ACM Press, 1986.

[12] D. Kempe, J. M. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. In *ICALP*, pages 1127–1138, 2005.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[14] J. M. Kleinberg. Bursty and hierarchical structure in streams. In *KDD*, pages 91–101, 2002.

[15] J. M. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639, 2002.

[16] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks*, 31(11-16):1481–1493, 1999.

[17] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. Tseng. Discovery of blog communities based on mutual awareness. In *3rd Annual Workshop on the Weblogging Ecosystem*, 2006.

[18] Y.-R. Lin, H. Sundaram, Y. Chi, J. Tatemura, and B. L. Tseng. Blog community discovery and evolution based on mutual awareness expansion. In *Web Intelligence*, pages 48–56, 2007.

[19] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[20] F. C. N. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *ACL*, pages 183–190, 1993.

[21] M. F. Porter. An algorithm for suffix stripping. *Program*, pages 387–397, 1980.

[22] S.-W. Son, H. Jeong, and J. D. Noh. Random field ising model and community structure in complex networks. *The European Physical Journal B*, 50:431, 2006.

[23] A. Strehl. *Relationship-based Clustering and Cluster Ensembles for High-Dimensional Data Mining*. PhD thesis, University of Texas at Austin, 2002.

[24] J. A. Tomlin. A new paradigm for ranking pages on the world wide web. In *WWW*, pages 350–355, 2003.

[25] G. Xu and W.-Y. Ma. Building implicit links from content for forum search. In *SIGIR*, pages 300–307, 2006.

[26] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML '97*, pages 412–420, 1997.

[27] H. Zhang, C. L. Giles, H. C. Foley, and J. Yen. Probabilistic community discovery using hierarchical latent gaussian mixture model. In *AAAI*, pages 663–668, 2007.