Topical Analysis for Identification of Web Communities

Yajie Miao and Chunping Li

Tsinghua National Laboratory for Information Science and Technology (TNList)
Key Laboratory for Information System Security, Ministry of Education
School of Software, Tsinghua University
Beijing, China 100084
miaoyj08@mails.tsinghua.edu.cn, cli@tsinghua.edu.cn

Abstract. Traditional link-based schemes for identification of web communities focus on partitioning the web graph more sophisticatedly, without concerning the topical information inherently held by web pages. In this paper, we give a novel method of measuring the topicality of a hyperlink according to its context. Based on this, we propose a topical maxflow-mincut algorithm which incorporates topical information into the traditional maxflow-mincut algorithm. Experiments show that our algorithm outperforms the traditional algorithm in identifying high-quality web communities.

Keywords: Link Analysis, Topical Analysis, Web Community, Web Structure Mining.

1 Introduction

The vast amount of information pertaining to various topics causes difficulties to web users in finding useful web pages while surfing on the net. To solve such a problem, researchers have been trying to reorganize the web in the form of communities, each of which is related with a single or several related topics. However, it is not an easy task to discover communities on the World Wide Web. Some link analysis based approaches to identification of web communities have been proved to be effective in some cases, one of the most well-known is the *maxflow-mincut* algorithm proposed in [12]. However, this “links-only” method still has its defects because it only utilizes link information and ignores contents of web pages.

In recent years, much attention has been paid to combination of link analysis with topical information. K. Bharat [7] regulated hub and authority scores calculated by HITS [1] using relevance between pages and topics. S. Chakrabarti gave each hyperlink in HITS a weight determined by the similarity of its anchor text with the topic. Topic-Sensitive PageRank [6] and Topical PageRank [3] improved PageRank from a topical perspective and obtained better rank results. However, none of these methods have attempted to combine topical analysis with identification of web communities.

In this paper, we suggest that the weight of a link should embody its topical context. Topical information is incorporated into the traditional method through *Topical Weight* and a *topical maxflow-mincut* algorithm is proposed. Experimental

A. Nicholson and X. Li (Eds.): AI 2009, LNAI 5866, pp. 101–110, 2009.

results show that significant improvements are achieved when using our proposed *topical maxflow-mincut* algorithm.

The remainder of this paper is organized as follows. Related work is introduced in Section 2. The traditional *maxflow-mincut* algorithm is described in Section 3. We present the *topical maxflow-mincut* algorithm in Section 4. In Section 5 the experiment and performance evaluation will be given. We have the concluding remarks and future work in Section 6.

2 Related Work

2.1 Hyperlink Weight

In the early literatures about HITS [1] and PageRank [2], the weight of each link was assumed to be 1 and all the links were treated uniformly. A new metric called *average-click* [5] was then introduced on the basic intuition that users would make a greater effort to find and follow a link among a large number of links than a link among only a couple of links. The weight of a link pointing from p to q was defined to be the probability for a "random surfer" to reach q from p, and therefore was determined by the number of outgoing links of p.

In [4], Chakrabarti et al pointed out that a hyperlink would be more important to the web surfer if the page it pointed to was relevant to this surfer's topic. In order to measure a hyperlink according to a specific topic, they examined the anchor text of a hyperlink and calculated its similarity to the descriptions of the topic. This text similarity was considered as the topical weight for this link.

2.2 Incorporating Topicality in Link Analysis

Bharat et al [7] defined a relevance weight for each web graph node as the similarity of its document to the query topic. Then this weight was used to regulate each node's hub and authority scores computed by HITS. Their experiments showed that adding content analysis could provide appreciable improvements over the basic HITS.

In Havaliwala's Topic-Sensitive PageRank [6], some topics were selected from predefined categories. For each topic, a PageRank vector was computed. A topic-sensitive PageRank score for each page was finally computed by summing up elements of all the PageRank vectors pertaining to various topics. By adopting these topical analysis methods, they captured more accurately the importance of each page with respect to topics.

Nie et al [3] presented a more sophisticated method for incorporating topical information to both HITS and PageRank. For each page, they calculated a score vector to distinguish the contribution from different topics. Using a random walk model, they probabilistically combined page topic distribution with link structure.

Experiments showed that their method outperformed other approaches.

2.3 Identification of Web Communities

Identifying communities on the web is a traditional task for web mining, knowledge discovery, graph theory and so on. Gibson et al [9] defined web communities as a core of central authoritative pages interconnected by hub pages and HITS was used to identify the authorities and hubs which formed a tightly knit community.

Kumar [10] represented web communities with community cores, which were identified through a systematic process called *Trawling* during a web crawl. Also with the HITS approach, these cores were used to identify relevant communities iteratively.

Flake et al [12] defined a web community as a set of sites that have more links (in either direction) to members of the community than to non-members. They proposed a *maxflow-mincut* method for identifying web communities. More details about this method will be discussed further in Section 3.

Ino et al [11] introduced a stricter community definition and defined the *equivalence relation* between web pages. After all the equivalence relations have been determined, the web graph can be partitioned into groups using a hierarchical process.

Lee et al [16] proposed to use content similarity between pages to give nodes weights and build new implicit links between nodes in the graph. However, their work focused mainly on viral communities and failed to take topical information and analysis into consideration.

3 Maxflow-Mincut Framework

Flake et al [12] proposed an algorithm for community identification on the World Wide Web. Since our topical approach is mainly based on this method, in this section we elaborate this method and examine its procedures in detail.

The web can be modeled as a graph in which web pages are vertices and hyperlinks are edges. Flake et al defined a web community as a set of websites that have more links to members of the community than to non-members. Also, they devised an iterative algorithm to find web communities. In each iteration, four steps are taken as follows.

First, starting from a set of seed pages, a focus crawler initially proposed in [8] is used to get a number of web pages through a crawl of fixed depth.

Then, a web graph is constructed using these web pages as well as relationships between them. In common cases, these relationships are represented by an adjacency matrix.

In the third step, one of the simplest maximum flow algorithms, i.e., the shortest augmentation path algorithm is run on the web graph and the minimum cut is identified.

The final step involves removing all the edges on the minimum cut found in the third step. All the web vertices which are still reachable from the source form a web community. In the community, all the pages are ranked by the number of links each one has and the highest ranked non-seed web page is added to the seed set.

These four procedures iterate until the desired iteration number is reached. In this paper, we call this algorithm *basic maxflow-mincut*. An example for *basic maxflow-mincut* is shown in Fig. 2(a). The web graph is composed of 10 nodes, each of which is marked by an integer. The seed nodes are 1 and 2. For the limit of space, we do not show the source and sink nodes.

4 Topical Identification of Web Communities

In the following parts, we first define *Topical Weight* (abbreviated as *TW*) to measure the topicality of hyperlinks. Based on this weight metric, we improve the basic algorithm and propose a *topical maxflow-mincut* algorithm.

4.1 Topical Weight for Hyperlinks

Commonly, several seed pages are used for crawling the web for pages used in identification of web communities. We use the TextRank [15] method to automatically extract some keywords from these seed pages. Some noisy contents, like anchor texts and advertisements, are filtered from these seeds manually. All the lexical units, i.e., words in our application, are regarded as vertices in the graph and an edge between two words is assumed to exist in the graph if these two words co-occur in one sentence. The salience score for each word can be calculated through iterative computations similar with PageRank [2]. Using a score threshold, some most important keywords can be selected to form a set, which is denoted as *W*.

Let's assume that a surfer is browsing page *i* which has a set of outgoing links *O*(*i*). A link *link*(*i, j*) is point from page *i* to page *j*. The more interesting page *j* is to the surfer, the more likely he or she is to follow *link*(*i, j*) for the next move. So the weight of a hyperlink can be measured by the interestingness of the page it is pointing to.

Since *W* can be viewed as a representative description of the topic, the interestingness of a page *P* can be measured by the number of keywords appearing in *P*. So the topical weight of the hyperlink pointing from page *i* to page *j*, denoted as *TW*(*i, j*), can be formally formulated as

$$TW(i, j) = \alpha \, Count(W, j), \tag{1}$$

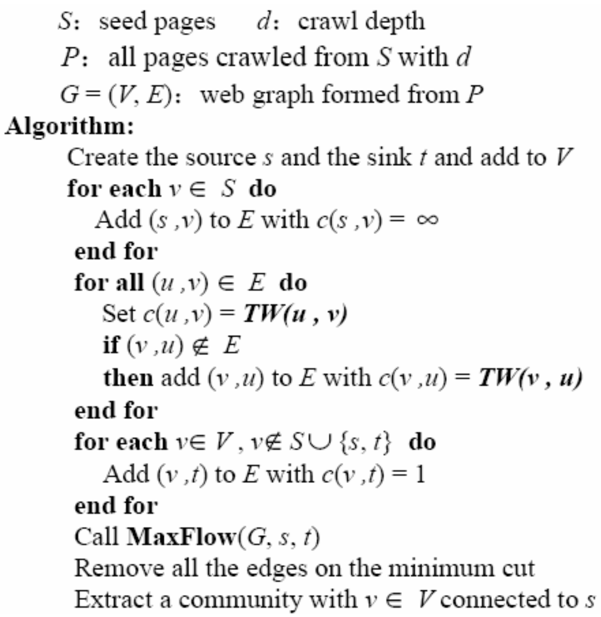
where *Count*(*W, j*) is the counting number of keywords appearing in page j and α is a regulating factor. For simplification, we set α as 1.

4.2 Topical Maxflow-Mincut

As discussed above, *basic maxflow-mincut* treats every link between nodes equally. We define *Topical Weight* for links and also give a feasible method for calculating a hyperlink's *Topical Weight*. Therefore, if we have an edge between vertices *u* and *v* in the web graph *G*=(*V*,*E*), where both *u* and *v* are neither the source *s* nor the sink *t*, we can use *TW*(*u, v*) as the weight for this edge. The *topical maxflow-mincut* algorithm is shown in Fig. 1.

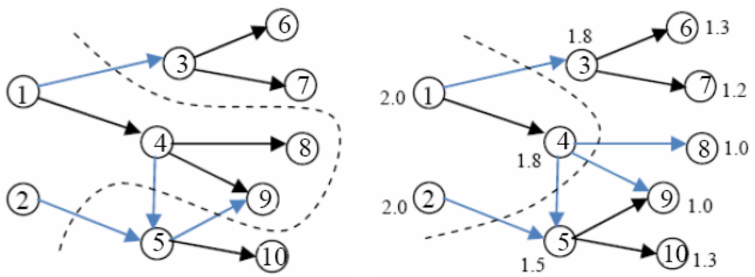
A subtle problem with this algorithm is that *TW(u, v)*, the keyword number, may have a very broad value spectrum. So we consider normalizing *Topical Weight* to a value in [*1.0*, β]. We set the lower bound of this normalization interval to be 1.0 because *Topical Weight* is required to be larger than the weight of links pointing from non-source and non-seed vertices to the sink. Suppose we have *TW* in an interval [*A*, *B*]. Using this normalization strategy, *TW* will be mapped to a value in [*1.0*, β] as

$$normalized\,(\mathrm{TW}) = 1.0 + \frac{TW - A}{B - A} \times (\beta - 1.0) \cdot \tag{2}$$



```
S: seed pages    d: crawl depth
P: all pages crawled from S with d
G = (V, E): web graph formed from P
Algorithm:
    Create the source $s$ and the sink $t$ and add to $V$
    for each $v \in S$ do
        Add $(s, v)$ to $E$ with $c(s, v) = \infty$
    end for
    for all $(u, v) \in E$ do
        Set $c(u, v) = TW(u, v)$
        if $(v, u) \notin E$
        then add $(v, u)$ to $E$ with $c(v, u) = TW(v, u)$
    end for
    for each $v \in V$, $v \notin S \cup \{s, t\}$ do
        Add $(v, t)$ to $E$ with $c(v, t) = 1$
    end for
    Call MaxFlow$(G, s, t)$
    Remove all the edges on the minimum cut
    Extract a community with $v \in V$ connected to $s$
```

Fig. 1. Topical maxflow-mincut algorithm



(a)

(b)

Fig. 2. Examples for both basic and topical maxflow-mincut algorithm. (a) An example for basic maxflow-mincut algorithm. (b) An example for topical maxflow-mincut algorithm.

We adopt this normalization operation and use normalized *Topical Weight* as the weight for hyperlinks. The upper bound β is a vital factor for our algorithm and this parameter will be tuned systematically to make the algorithm achieve its best performance.

In Fig. 2(b), the nodes are given different weights showing their varied relevance to the topic. These weights have a range [1.0, 2.0], where 1.0 means “totally irrelevant” and 2.0 means “perfectly relevant”. So each edge in the graph also obtains accordingly a weight equal to the weight of the node it is pointing to. We identify a community with *topical maxflow-mincut* algorithm on this web graph.

Similarly, the minimum cut is marked with the dashed curve and the edges in blue are removed. Compared with Fig. 2(a), a community consisting of three members, 1, 2, 4, is identified, excluding two totally irrelevant nodes, 8 and 9. This illustrates the superiority of our algorithm: by utilizing additional topical information, the algorithm can eliminate irrelevant or noise nodes and generate more desirable communities.

5 Experiment and Evaluation

5.1 Metrics for Measuring Web Communities

In order to make our evaluation at a quantitative level, we define some metrics for measuring the quality of web communities. Here are several page sets:

- W : the set of all the pages generated through crawling
- C : the set of pages in the community generated on W
- R : the set of pages which are included in W but excluded from C

So these three page sets have such properties:

$$C \in W \quad R \in W \quad R = W - C. \tag{3}$$

We examine both *C* and *R*, and record the number of relevant and irrelevant pages. Suppose after manually counting, we discover *a* relevant pages and *b* irrelevant pages in *C*. Similarly, *c* relevant pages and *d* irrelevant pages are discovered in *R*. We define *Purity* as the proportion of relevant pages in *C* to the total number of pages in *C* and *Coverage* as the proportion of relevant pages in *C* to the total number of relevant pages in *W*. Formally, *Purity* and *Coverage* can be defined as

$$Purity(C) = a/(a+b) \quad Coverage(C) = a/(a+c). \tag{4}$$

Purity measures how pure a community is and *Coverage* evaluates how much the community can cover the relevant pages in the whole page set *W*. From their definitions, we can see that either *Purity* or *Coverage* just shows one aspect of a community. In order to avoid our evaluation being biased towards a single aspect, we combine *Purity* and *Coverage*, and define *F-measure* in a usual way

$$F-measure = \frac{2 \times Purity\ (C) \times Coverage\ (C)}{Purity\ (C) + Coverage\ (C)} \tag{5}$$

5.2 Data Set

We select the "data mining conferences" community as our testing case. Our seed set consists of three URLs:

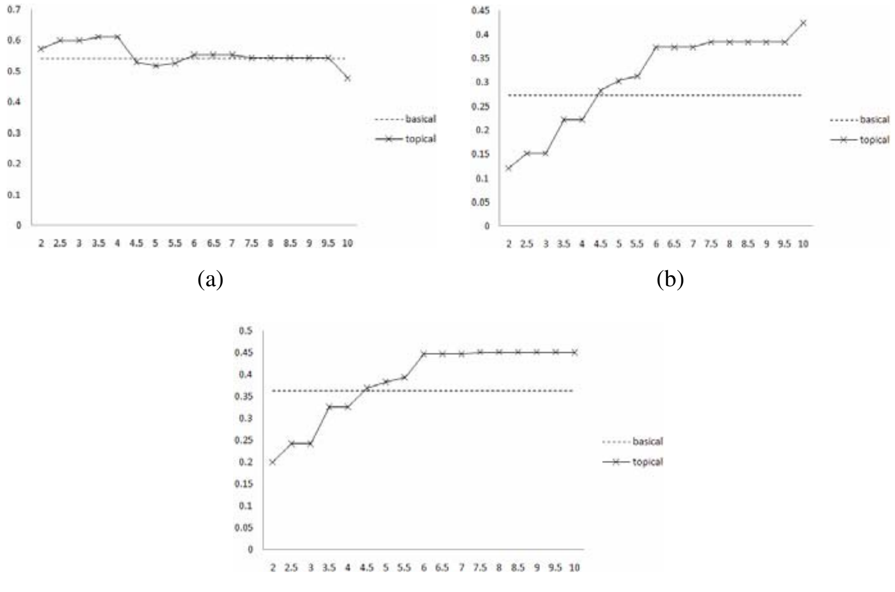
http://www.kdnuggets.com/meetings/
http://www.sigkdd.org/kdd2009/
http://www.kmining.com/info_conferences.html

From the contents of these three pages, we extract 39 keywords, which will be used to calculate *Topical Weight* for edges in the web graph. These keywords include terminologies on data mining, names of academic organizations, well-known data mining conferences, different types of conferences like “workshop”, “symposium”, “conference”, etc., as well as terms frequently used for conference affairs like “registration”, “acceptance” and “notification”, etc.

After crawling, we totally get 389 distinct pages as the data set. Similar to [3], we rate manually each page as quite relevant, relevant, not relevant, and totally irrelevant, which is assigned the scores of 4, 3, 2 and 1, respectively. We mark pages with scores of 4 or 3 as relevant and pages with scores of 2 or 1 as irrelevant. One point which should be noted is that we are especially interested in a methodology. Therefore, we use a much smaller dataset in our experiment, which makes it feasible for manual rating and close examination of score distributions.

5.3 Parameter Tuning

The parameter β is the upper bound of the normalization interval [*1.0*, β]. We tune β with values from 2.0 to 10.0 with a step of 0.5. For each value of β, we run *topical maxflow-mincut* algorithm and generate a web community C_β. For C_β we calculate its *Purity*, *Coverage* and *F-measure* according to their definitions. Fig. 3 shows the values of these metrics with different settings of β. We depict the value level of *basic maxflow-mincut* with the dashed line.



(a)

(b)

(c)

Fig. 3. Values of metrics as β is varied. (a) Community Purity as β changes from 2.0 to 10.0. (b) Community Coverage as β changes from 2.0 to 10.0. (c) Community F-measure as β changes from 2.0 to 10.0.

Fig. 3(a) demonstrates that with β=4.0, *Purity* can get its largest value, which approximately is 0.62 and gains 15% improvements against the basic algorithm. From Fig. 3(c) we can see that when β equals 6.0, the *F-measure* curve reaches its highest point. When β is larger than 6.0, *F-measure* stays unchanged, confirming that increasing β continuously cannot boost the performance anymore. Fig. 3(a) and Fig. 3(b) show that when β is 6.0, both *Purity* and *Coverage* are larger than that of the basic algorithm, though for *Purity* the improvements are not significant. With β=6.0, we get an community containing 66 members, an appropriate size considering the total number of the pages in the data set is 389.

As β varies, *Purity* displays an opposite changing tendency to that of *Coverage*. For a specific application, it is almost impossible to optimize these two competing metrics simultaneously. This demonstrates that both *Purity* and *Coverage* are partial in measuring a community. Therefore it is more reasonable to use *F-measure* as our major evaluation metric.

5.4 Performance Evaluation

Table 1 shows performance comparisons between topical and basic *maxflow-mincut* algorithms. Since *F-measure* should be our major evaluation metric, we can conclude that *topical maxflow-mincut* improves the performance of *basic maxflow-mincut* by 23.041%. However an observation from Table 1 is that *topical maxflow-mincut* improves *Purity* by less than 0.1%. Since *F-measure* is a combination of *Purity* with *Coverage*, the appreciable improvements on *F-measure* mainly come from *Coverage*. It appears that *topical maxflow-mincut* improves *basic maxflow-mincut* only by expanding the community and as a result increasing the value of *Coverage*. Next we will prove that besides enlarging the size of the identified community, our algorithm indeed improves the topicality quality of the community.

Table 1. Performance comparison with β=6.0

Metric	Topical	Basic	Improvement
Purity	0.5523	0.5400	0.023%
Coverage	0.3738	0.2727	37.074%
F-measure	0.4459	0.3624	23.041%

We have manually rated all the pages with scores ranging from 1 to 4. In order to provide an insight into the score distribution among the community members, we rank them into a list and calculate the overall average score. In [12], each page in the community was assigned a score equal to the sum of the number of its inbound and outbound links and all the pages were ranked according to their scores. For *basic maxflow-mincut*, we take the same ranking scheme. But for *topical maxflow-mincut*, the score of each page is the sum of topical weight of its inbound and outbound links. We define *S@n* as the average score of the first *n* pages in the ranking result. As the size of the community formed by *basic maxflow-mincut* is 49, we set *n* to be 5, 10, 15, 20, 25, 30, 35, 40, 45, 49, and for each value we calculate *S@n* correspondingly for both *topical* and *basic maxflow-mincut.* A comparison is made in Fig. 4. For most

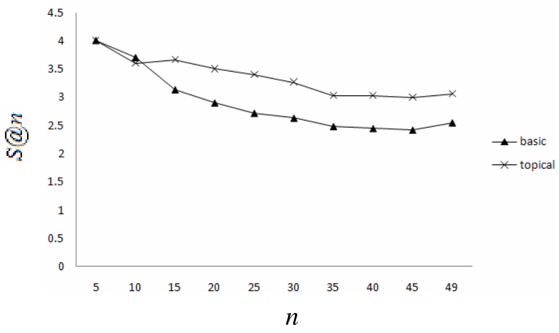


Fig. 4. Comparison of average score as n is varied

values of *n*, *S@n* of *topical maxflow-mincut* is higher than that of *basic maxflow-mincut*. This superiority is retained until the last member of the basic community. So we can conclude that besides expanding the community to improve *F-measure*, *topical maxflow-mincut* pulls more high-quality and authoritative pages into the community as well. Also Fig. 4 shows that another advantage of *Topical Weight* is that it can be used as a more effective metric in ranking members of a web community as it can give relevant pages higher positions on the ranking list.

6 Conclusion and Future Work

In this paper, we make a preliminary attempt to incorporate topical analysis into identification of web communities. We use *Topical Weight* to measure the topicality of hyperlinks. By combining *Topical Weight* with the traditional *basic maxflow-mincut* algorithm, we propose *topical maxflow-mincut*, which is an improved algorithm for identification of web communities. Also we define some metrics for measuring the quality of web communities. Experimental results show that our algorithm achieves improvements over *basic maxflow-mincut* and is more capable of finding high-quality web communities.

In our future work, we expect to capture the topicality of hyperlinks more accurately using other methods like TF-IDF or Topic Signature. We would also consider incorporating other types of information, like opinions expressed in the contents of web pages, to the analysis of web communities, which may make it possible for us to identify communities with sentiment characteristics.

Acknowledgments

This work was supported by National Natural Science Funding of China under Grant No. 90718022.

References

1. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
2. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the 7th International World Wide Web Conference, pp. 107–117 (1998)
3. Nie, L., Davison, B.D., Qi, X.: Topical link analysis for web search. In: Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval, pp. 91–98 (2006)
4. Chakrabarti, S., Dom, B.E., Raghavan, P., Rajagopalan, S., Gibson, D., Kleinberg, J.M.: Automatic resource compilation by analyzing hyperlink structure and associated text. In: Proceedings of the 7th International World Wide Web Conference, pp. 65–74 (1998)
5. Matsuo, Y., Ohsawa, Y., Ishizuka, M.: Average-clicks: a new measure of distance on the World Wide Web. Journal of Intelligent Information Systems, 51–62 (2003)
6. Haveliwala, T.H.: Topic-sensitive PageRank. In: Proceedings of the 11th International World Wide Web Conference, pp. 517–526 (2002)
7. Bharat, K., Henzinger, M.R.: Improved algorithms for topic distillation in hyperlinked environments. In: Proceedings of the 21st International ACM SIGIR conference on research and development in information retrieval, pp. 104–111 (1998)
8. Chakrabarti, S., van der Berg, M., Dom, B.: Focused crawling: a new approach to topic-specific web resource discovery. In: Proceedings of the 8th International World Wide Web Conference (1999)
9. Gibson, D., Klienberg, J., Raghavan, P.: Inferring web communities from link topology. In: Proceedings of the 9th ACM conference on hypertext and hypermedia, pp. 225–234 (1998)
10. Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.: Trawling the web for emerging cyber-communities. In: Proceedings of the 8th International World Wide Web Conference, pp. 65–74 (1999)
11. Ino, H., Kudo, M., Nakamura, A.: Partitioning of web graphs by community topology. In: Proceedings of the 14th International World Wide Web Conference, pp. 661–669 (2005)
12. Flake, G.W., Lawrence, S., Giles, C.L.: Efficient identification of web communities. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 150–160 (2000)
13. Flake, G.W., Lawrence, S., Giles, C.L., Coetzee, F.M.: Self-Organization and identification of web communities. IEEE Computer 35(3), 66–71 (2002)
14. Page, L.: PageRank: Bringing order to the web. Stanford Digital Libraries Working Paper (1997)
15. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
16. Lee, H., Borodin, A., Goldsmith, L.: Extracting and Ranking Viral Communities Using Seeds and Content Similarity. In: Proceedings of the nineteenth ACM conference on hypertext and hypermedia, pp. 139–148 (2008)