Discovering Community Preference Influence Network by Social Network Opinion Posts Mining

Tamanna S. Mumu and Christie I. Ezeife*

School of Computer Science, University of Windsor, Windsor, Ontario N9B 3P4, Canada
{mumut,cezeife}@uwindsor.ca

Abstract. The popularity of posts, topics, and opinions on social media websites and the influence ability of users can be discovered by analyzing the responses of users (e.g., likes/dislikes, comments, ratings). Existing web opinion mining systems such as OpinionMiner is based on opinion text similarity scoring of users' review texts and product ratings to generate database table of features, functions and opinions mined through classification to identify arriving opinions as positive or negative on user-service networks or interest networks (e.g., Amazon.com). These systems are not directly applicable to user-user networks or friendship networks (e.g., Facebook.com) since they do not consider multiple posts on multiple products, users' relationships (such as influence), and diverse posts and comments. This paper proposes a new influence network (IN) generation algorithm (Opinion Based IN:OBIN) through opinion mining of friendship networks. OBIN mines opinions using extended OpinionMiner that considers multiple posts and relationships (influences) between users. Approach used includes frequent pattern mining algorithm for determining community (positive or negative) preferences for a given product as input to standard influence maximization algorithms like CELF for target marketing.

Keywords: Social network, Influence analysis, Sentiment classification, Recommendation, Ranking, Opinion mining.

1 Introduction

People may give their opinions more often on shared posts on social network where those opinions may be positive, negative, or controversial to the shared posts. At present, considering communities extracted from social graphs and monitoring the aggregate trends and opinions discovered by these communities has shown its potential for a number of business applications such as marketing intelligence and competitive intelligence. The tasks include identification of influential posts, influential persons and services, users' opinions analysis, and community detection based on shared interest. These tasks can be performed through data mining approaches which include classification, clustering, association rule mining, sequential pattern mining.

* This research was supported by the Natural Science and Engineering Research Council (NSERC) of Canada under an operating grant (OGP-0194134) and a University of Windsor grant.

L. Bellatreche and M.K. Mohania (Eds.): DaWaK 2014, LNCS 8646, pp. 136–145, 2014.

Social Network Data - A social network framework is generally represented as a graph $G(V, E)$, where V is the set of nodes representing users and E is the set of edges between nodes representing a specific type of interaction between them. The edges may be directed or undirected. Bonchi et al. [3] define the links between nodes in two ways. *Explicit* e.g., users declaring explicitly their friends or connections through such actions as joining a group, liking a page, following a user or accepting a friendship request. Another type of link is *Implicit* e.g., links identified from users' activities by analyzing broad and repeated interactions between users such as voting, sharing, tagging, or commenting. Viral marketing is an approach of information spreading, where a small set of influential customers can influence greater number of customers [2]. A major and common issue in the area of existing opinion mining is to identify product popularity based on one specific feature such as sentiment of comments [6], [4], or rating on topic [8]. However, in friendship networks (e.g., facebook.com), users' opinions on a product are defined implicitly or explicitly in the networks, and unlike in interest networks (e.g., Amazon.com), influence on a product not only depends on a specific product webpage but also on the complex relationships between users connected in the networks, and this requires all kinds of implicit and explicit opinions and relationships that need to be identified and aggregated.

In standard influence maximization (IM) systems such as CELF [7], the whole social network is taken as input to find influential users as seed set for a specific product (e.g., iPhone) for target marketing [2]. The main limitation of general IM systems like CELF is that they do not consider multiple posts on multiple products as well as relationships between users, and thus, are not effectively product-specific because of the need to first search large social network data for multiple influential users/product opinions who may not be influential on a specific product (e.g., iphone). So, considering those users as influential for a product reduces the accuracy and efficiency of such general IM algorithms.

1.1 Contributions and Outline

Motivated by the issues described above, the problem we tackle is as follows:

Problem Definition – Build an influential network (IN) generation model for influence maximization of a specific product based on mining users' posts and opinions (positive or negative) on a specific product and relationships from a friendship network graph $G(V, E)$ where every edge $e_{ij} \epsilon E$ connects nodes v_i and v_j ($v_i, v_j \epsilon V and\ i, j = 1,2, \dots, N$) and indicates v_i and v_j have relationships on a specific product. Also, measure influence acceptance score of each node v_i for a product and remove nodes that are below certain threshold before applying IM algorithm on the pruned product-specific friendship network to more effectively and efficiently compute a product-specific IM. To solve this problem, paper contributions are:

1) We propose a new influence network generation model for friendship networks by mining opinions named Opinion Based IN (OBIN) which incorporates implicit, explicit opinions and complex user-user and user-product relationships to get a ranked list of users, opinions and relationships in a Topic-Post Distribution (TPD) model.

2) Based on TPD model, we propose a new opinion mining framework named Post-Comment Polarity Miner (PCP-Miner) which is an extension of OpinionMiner [6] augmented with the Apriori [1] frequent pattern mining technique, that considers multiple posts, relationships between posts and non-tagged comments, and then generates pruned IN.

Section 2 of this paper presents related work, section 3, the proposed system, the OBIN. Section 4 the experimental results, and section 5 conclusions and future works.

2 Related Work

Recent research on Social Network have analyzed social network data to find pattern of popularity or influence in various domains such as blogging (e.g., Slashdot.org), micro-blogging (e.g., Twitter.com), bookmarking (e.g., Digg.com), co-authorship (e.g., Academia.edu), movie review (e.g., IMDb.com), and product review domains (e.g., Amazon.com). Our proposed work is motivated by [6] in product review domains; and the work of [7] and [2] in influence maximization.

Dave et al. [4] proposed an opinion extraction and mining method based on features and scoring matrices and classified review sentences as positive or negative. Hu and Liu [5] proposed a feature-based summarization (FBS) that mines product features from customers' reviews. Jin et al. [6] also worked similar as [5] and defined four entity types, eight tags and four pattern tag sets to the feature-based approach, named OpinionMiner. But they ignore opinions that contain different product information and infrequent product features. Existing influence maximization algorithms such as Cost-Effective Lazy Forward selection (CELF) [7] require that the influence probability is known and given to the algorithm as input along with a social network graph. Trust-General Threshold (TGT) model proposed by [2] works in trust network considering trust and distrust of nodes and improved [7]. Both CELF and TGT do not consider user-user relationships that can be obtained from friendship networks such as Facebook. and product specific problems. TGT also requires trust and distrust to be explicitly described.

In this paper, we propose a product-specific influence network generation model based on users' opinions on friendship network to discover community preference of relevant users by considering implicit and explicit users relationships, which shows target marketing is more focused than [7] and [2] with the benefit of computing a more accurate influential network for a product.

3 The Proposed OBIN Model

The proposed OBIN model takes a social network graph $G(V, E)$ and a product z as input and generates an influence graph $Gz(V, E)$ on product z from computed community preference where V is the relevant nodes extracted from G. Our proposed pruning strategy is based on number of nodes, likes and shares, number of positive and negative comments, and extracted user-user relationships. Fig. 1 shows the algorithm for OBIN model. OBIN has 3 main functions, TPD (Topic-Post Distribution) (lines 1-4 in Fig. 1), PCP-Miner (Post-Comment Polarity Miner) (lines 5-7 in Fig. 1), and influence network generator (line 8 in Fig. 1).

3.1 Topic – Post Distribution (TPD) Model

For a given product z, our proposed TPD extracts all nodes, posts, and comments by applying SQL queries to friendship network graph, to mine the dataset to classify relevant and irrelevant nodes of product z, determined by the number of nodes connected to influential node and number of likes on the posts denoted by $A(Approve)$, number of shares and comments on the posts denoted by $SR(Simple\ Response)$, product information as search key (e.g., iPhone screen resolution) denoted by $Term$. We compute $Approve\ A$ in two levels, (1) in the node selection step A_f represents the number of friends connected to the node, and (2) in the post selection step A_l represents the number of likes on the post.. The collective approved rating of a node V is the sum of the two approved ratings as $A(V) = A_f + A_l$.

Algorithm OBIN to generate influence network graph G_z from friendship network G
Input: Social network URL (e.g., facebook.com), product z, Approve A, Simple response SR, $Term$ in product name z
Output: Set of influential nodes V_s, influenced nodes V_t, influence graph G_z on z

---

1. *Execute SQL query on URL to find set of nodes on product z using Graph API*
2. *Generate nodes matrix NM with 4 attributes* $< node, Term, A, SR >$
3. *Generate relevant nodes matrix PM with 4 attributes* $< node\ (V),\ Term,\ A,\ SR >$ *by* ***mining*** NM *with three features* $(Term + A)$, $(Term + SR)$, $(A + SR)$, *to classify relevant and irrelevant nodes using SVM classifier.*
4. *Execute SQL query to find set of posts and comments of* V *from PM. Store posts* w *in table tblPosts and comments* c *in table tblComments.*
5. ***FOR*** *each comment* c *in table tblComments* ***DO***
   - 5.1. *Execute tokenization and POS-tagging process as described in section 3.2*
   - 5.2. *Generate* $FeqFT\ (c, FFT)$ *matrix by identifying frequent features (FFT) in* c *through* ***Apriori*** *frequent pattern algorithm with minimum support 50%*
   - 5.3. *Identify opinion words* OW *for extracted* FFT *as described in section 3.2*
   - 5.4. *Determine semantic orientation* OR *of* OW *as described in section 3.2*
   - 5.5. *Generate* $OE\ (c, FFT, OW, OR)$ *matrix of* c
   - 5.6. *Store node* $V_t \in V$, *who commented c, in the matrix named* VT*matrix (influenced nodes matrix)*
6. ***FOR*** *each post W in table tblPosts* ***DO***
   - 6.1. *Compute the polarity score* θ_Z *from* OE *matrix as* $\theta_Z = (\sum positive\ c - \sum negative\ c) \times 100\%$
   - 6.2. *Store node* $V_s \in V$, *who posted* W, *in the matrix named* VS *matrix (influential nodes matrix)*
7. *Merge* VT *and* VS *matrices into influence matrix IMAT with 3 attributes* $< influencial\ users(V_s), influenced\ users(V_t), Action >$ *as follows:*
   - 7.1. ***IF*** *node* V_t *responds to node* V_S ***Do*** $IMAT[Action] = \sum responses$
   - 7.2. ***ELSE Do*** $IMAT[Action] = 0$
8. *Generate influence graph* $G_z = (V, E)$, $V \in VT, VS$ *and* $E = IMAT[Action]$ *if there exist a relationship between* VT *and* VS *matrices (section 3.3)*

Fig. 1. Algorithm of OBIN to generate influential network from friendship network

TPD consists of three steps: relevant nodes identification (line 1 in Fig. 1), preprocessing (lines 2-3 in Fig. 1), and extraction (line 4 in Fig. 1). Identification is done by conducting a local search in the whole social network using Graph API (to crawl social network) and SQL. Let us consider the social network as Facebook.com and for a given product z, search using Graph API and FQL (Facebook SQL) results in a set of relevant nodes V (users) on z as $< V, Term, Approve, Link >$ where $Link$ denotes profile URL of a user (V). To determine if node V has any influence on z, we need a node approval threshold Az determined by the user based on the following logic, and this can be adjusted after various runs to get desired ranges of thresholds. The higher the threshold, the higher the influence spread of selected relevant nodes. If $A(V) \geq Az$, TPD extracts all the relevant posts posted by V on z as $< W, Term, A, SR >$, where W is the published post, SR is the summation of comments and shares of W.

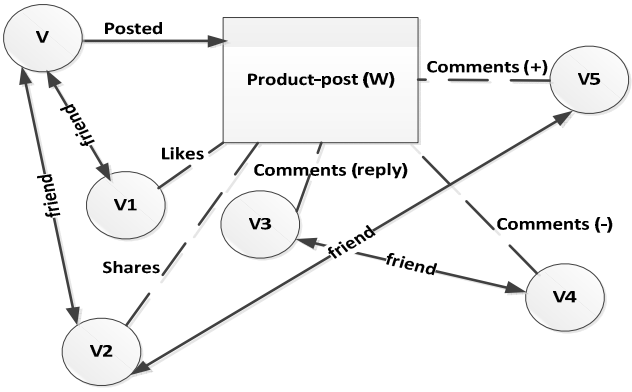

Fig. 2. An example activity in friendship network

Fig. 2 shows an example of activities in a friendship network where V posted a post (W) on a product, and $V1$, $V2$, $V3$, $V4$, $V5$ have expressed their opinions in different ways. For example, if we decide $Az = 50$, then TPD will extract all posts published by V on z having ($number\ of\ likes \geq 50$). To extract all the possible relevant posts, TPD mines the dataset with three different search features such as search term plus number of approval ($Term + A$), term plus number of simple responses ($Term + SR$), and sum of approval and simple responses ($A + SR$), to classify relevant and irrelevant nodes. A profile d can be denoted as a vector of N posts W_N and node V. For N number of posts and i number of relevant nodes, we denote the profile documents as D: $\{ d1, d2, d3, \ldots, di \}$, di: $\{(W_1, Vi), (W_2, Vi), \ldots, (W_N, Vi)\}$. TPD keeps track of $V \times D$ (nodes by profiles) matrix, $D \times W$ (profiles by posts) matrix, and $W \times C$ (posts by comments) matrix.

For example, if we execute the query "*SELECT id, name, category, likes, link FROM search WHERE q = 'iPhone' AND (type = 'page' OR type = 'group')*", then TPD will produce a set of relevant nodes on 'iPhone'. Table 1 shows an example of extracted relevant nodes. After indexing and applying threshold, if we execute a query

such as "*SELECT post_id, message, likes.count AS A, share_count, created_time, comments.count, (comments.count+share_count) AS SR FROM stream WHERE source_id = '1' AND message !='' ORDER BY likes.count LIMIT 100*", then TPD produces a set of 100 posts on z for node id = *1*. Table 2 and 3 show examples of extracted data by TPD from social network.

Table 1. Example of extracted nodes by TPD

Node ID Vs	Term	Approve A	Link
1	iphone	3116728	iphone.page
11	iphone 4	1435239	Iphone-4

Table 2. Example of extracted posts by TPD

Post ID W	Term	Approve A	Simple response SR
46947	Black or white	61153	11325

Table 3. Example of extracted comment data by TPD with node ID who commented

Post ID W	User ID Vt	Time	Comment C
46947	108936	2013-01-06	this is really cool

3.2 Post – Comment Polarity Miner (PCP – Miner)

In a friendship network, users are free to comment on any published post to express their opinions. TPD gives a ranked list of relevant nodes ($\boldsymbol{V}$), posts ($\boldsymbol{W}$) and comments ($\boldsymbol{C}$). To determine the influential capability of a node $\boldsymbol{V}$, we need to compute the polarity score ($\boldsymbol{\theta_Z}$) of each post published by $\boldsymbol{V}$. Proposed PCP-Miner identifies opinion comments among all the comments, identifies semantic orientation ($\boldsymbol{SO}$) of the comments, and measures the polarity score ($\boldsymbol{\theta_Z}$) of the posts.

Step1: In our proposed method, data cleaning includes removal of stopwords, stemming, and fuzzy matching to deal with word variations and misspelling, fuzzy duplication removal, removal of comments that are not exact replicas but exhibit slight differences in the individual data values extracted from the same node, removal of suspicious links to protect spam. The PCP-Miner then uses Tokenization and part-of-speech tagger[9] to identify phrases in the input text.

Step 2: Identify product features on which many people have expressed their opinions. In this paper, we improve existing opinion mining system OpinionMiner by identifying explicit and implicit features. To identify implicit features, we have added Apriori frequent pattern to extract frequent features. For example, in the case of "iPhone", some users may use "resolution" as a feature for "camera", some use as "screen", some use as "video call", as shown in Table 4. To identify which itemsets are product features, we use Association rule mining [1] to identify frequent itemsets, because those itemsets are likely to be product features. In our work, we define an itemset as frequent if it appears more than 50% in the review set, so as not want to lose any important comment while avoiding spam comments.

Table 4. Example of frequent features

Comment Id	Features
1	Camera {resolution, camera}
2	Picture {resolution, camera, picture}
3	Screen resolution {resolution, screen}
4	Picture {resolution, video_call, camera, picture}

The input to the Apriori algorithm is the set of nouns or noun phrases from POS-tagging, and the output itemset is product features. **Association rule mining** – In the above example, frequently occurred items such as *resolution*, *camera*, and *picture* may lead to finding association rules *resolution => camera*, which means that the comment that mentions *resolution*, may usually refer to the feature *camera*. Here, the set {*resolution, camera*}, {*resolution, camera, picture*}, {*resolution, screen*}, and {*resolution, video_call, camera, picture*} are called itemsets. Now suppose, $resolution = 30$ (the number of comments mentioning resolution), $camera = 20$ (the number of comments mentioning camera), $both = 20$ (the number of comments mentioning both resolution and camera), and $total = 100$ (the number of comments on the post). So $support = {|Rule|}/{|total|} = \frac{20}{100} = 20\%$. Rules that satisfy a minimum support, are called frequent features.

Step 3: Extract opinion words and their semantic orientations. For example, "This picture is awesome" has the word 'awesome' is the effective opinion of 'picture' and it has positive orientation. Presence of adjectives in comment text is useful for predicting whether a text is expressing opinion or not. In the opinion words extraction phase, our proposed method takes the list of tokens with corresponding POS-tags [9] from our transactional database, and search for whether it contains adjective words and/or frequent features identified by Apriori algorithm. In our proposed work, we use WordNet to get the adjective synonym (words with similar meaning) set and antonym (words with opposite meaning) set to identify the opinion expressed by the word (i.e., positive or negative opinion). To identify the semantic orientation(positive/negative) of each comment, we need to identify the semantic orientation of extracted opinion words. This is done with the help of WordNet, we store some known orientations along with the words. When we will extract any new unknown word that is not in the list of WordNet, we look for its synonym and consider its' semantic orientation as the orientation of unknown word, and store it back into the list of semantic orientations for future use. For example, we know the word "good" has positive orientation stored in the list and has a list of synonyms {great, cool, awesome, nice}. If we find a comment containing the word 'cool', we will take its orientation positive. If a comment sentence contains a set of features, then for each feature, we compute an orientation score for the feature. Positive opinion word has score $(+1)$ and negative opinion word has score (-1). All the scores are then summed up. If the final score is positive, the semantic orientation of the comment is positive otherwise negative and if the score is zero then neutral.

Step 4: We calculate the polarity score θz of each post as

$$\theta_Z = \left(\sum positive\ responses - \sum negative\ responses\right) \times 100\% \quad (1)$$

3.3 Social Influence Graph Generation and Community Preference

Based on polarity score, we have a ranked list of relevant nodes $\boldsymbol{V}$, their posts $\boldsymbol{W}$, comments $\boldsymbol{C}$, and the set of influenced nodes who commented on the posts $\boldsymbol{W}$. From the set of influenced nodes, we compute the influence score determined by their number of responses and then index them. Table 6 (for example) shows a list of influenced nodes. We propose an algorithm PoPGen (popularity graph generator) to generate a social network influence graph $\boldsymbol{G_z = (V, E)}$ on product $\boldsymbol{z}$ using the influence matrix $\boldsymbol{IMAT}$ (Table 7). PoPGen adds a node $\boldsymbol{V}$ to the vertex list according to the number of responses. For all vertices, PoPGen finds if $\boldsymbol{V_i}$ has a relation with $\boldsymbol{V_j}$ where $\boldsymbol{V_i}$, $\boldsymbol{V_j \in V\ and\ i, j \in N(number\ of\ nodes\ in\ network)}$, add 1 in Table 7, for example, and $\boldsymbol{0}$ otherwise. The value 1 means add an edge between the vertices $\boldsymbol{V_i}$ and $\boldsymbol{V_j}$. The generated influence graph *Gz* represents the community preference for a product *z*.

Table 5. Example data for post – user relationship and user – user relationship

Node id V_i	Post ID W	Node id V_j
1	49823667	4
2	11250901	6

User id V_1	User id V_2
3	1

Table 6. Example data for Influence Matrix (IMAT) according to Table 6

	1	**2**	**3**	**4**	**5**	**6**	**7**
1	0	0	1	1	0	0	0
2	0	0	0	0	0	1	0

4 Experimental Evaluation

4.1 Dataset

We conducted our experiment using the users' posts and opinions in Facebook as a friendship network for iphone, iPad, Samsung Galaxy. Our TPD method automatically extracts the relevant data and stored into our generated data warehouse, called OBIN_dwh, in a temporal basis. We selected a set of datasets based on Approve ($\boldsymbol{A}$) and Simple response ($\boldsymbol{SR}$) as follows:

(1) Node selection – we considered $\boldsymbol{A_f \geq 1000}$ i.e., nodes having more than $\boldsymbol{1000}$ friends. With this characteristic, we had $\boldsymbol{1178}$ nodes with $\boldsymbol{42,664}$ relevant and irrelevant posts. (2) Post selection – we considered $\boldsymbol{SR \geq 20}$ and $\boldsymbol{A_l \geq 10}$ for

posts of each node where $\boldsymbol{SR}$ represents number of re-shares and comments of the posts and $\boldsymbol{A}$ represents number of likes and the result is $\mathbf{3793}$ relevant posts.

After applying TPD and PCP-Miner, we obtained $\mathbf{343}$ influential nodes and $\mathbf{45,126}$ influenced nodes with $\mathbf{47,298}$ relationship edges.

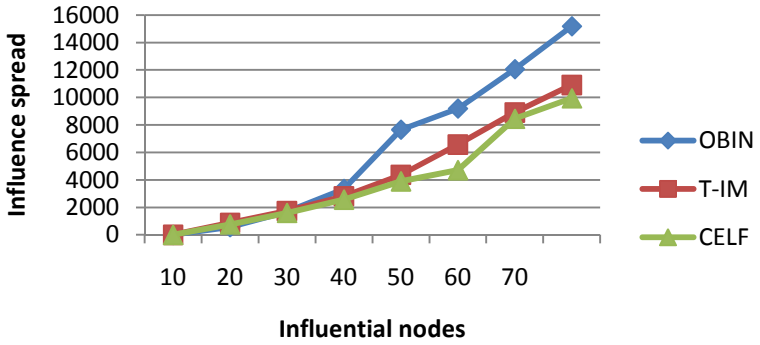

Fig. 3. Comparison of influence spread by different number of influential nodes

4.2 Performance Analysis

We evaluated proposed OBIN against accuracy of CELF and T-IM algorithms. **Recall –** is the ratio of the number of relevant nodes retrieved and the total number of relevant nodes that exist in the network, denoted by R. **Precision –** is the ratio of the number of relevant nodes retrieved and the total number of relevant and irrelevant nodes retrieved, denoted by P. **F-score –** accuracy of the proposed system $= 2 \times (P \times R)/(P + R)$. Table 8 shows the accuracy measure of CELF [7] and T-IM [2] and proposed OBIN with the same dataset, and we observed that OBIN is dramatically better by retrieving more relevant nodes than CELF and T-IM.

Table 7. Comparison of discovering influential nodes by CELF, T-IM and OBIN

	CELF	T-IM	OBIN
F – score	85.4%	88.1%	95.3%

Fig 3 shows the influence spread over network by different algorithms. With small number of nodes, CELF and T-IM and proposed OBIN give almost the same performance in influence spread, but as we increase the number of nodes, OBIN performs better because, for a specific product, CELF and T-IM discover relevant nodes along with more irrelevant nodes which slow down their performances.

5 Conclusions and Future Works

This paper proposed an effective method for discovering relevant influential nodes from friendship network which enables more focused target marketing than existing influential maximization algorithms. Previous research considers opinion mining only in user-service network that are not directly applicable to user-user network i.e., friendship network, where user-user network is a more complex network that includes multiple relationships between users and users, and between users and products. The proposed OBIN miner mines opinions from complex user-user relationship network (e.g., Facebook) with multiple posts, multiple products, considering both implicit and explicit opinions. Experimental results show that the proposed technique performs better than the existing general IM methods. To handle more rapid network evolution, future work will look into using pre-updating of data such as friendship relationships to speed up the computation of the IN by the OBIN system.

References

1. Agrawal, R., Srikant, R., et al.: Fast algorithms for mining association rules. In: 20th Int. Conf. Very Large Data Bases, VLDB, pp. 487–499 (1994)
2. Ahmed, S., Ezeife, C.I.: Discovering Influential Nodes from Trust Network. In: ACM SAC International Conference, Coimbra, Portugal (2013)
3. Bonchi, F., Castillo, C., Gionis, A., Jaimes, A.: Social network analysis and mining for business applications. ACM Transaction TIST 22 (2011)
4. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: 12th International Conference on World Wide Web, pp. 519–528 (2003)
5. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: 10th ACM SIGKDD Int. Conference on Knowl. Discov. and Data Mining, pp. 168–177 (2004)
6. Jin, W., Ho, H.H., Srihari, R.K.: OpinionMiner: a novel machine learning system for web opinion mining and extraction. In: 15th ACM SIGKDD Int. Conference on Knowl. Discov. and Data Mining, pp. 1195–1204 (2009)
7. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: 13th ACM SIGKDD Int. Conference on Knowl. Discov. and Data Mining, pp. 420–429 (2007)
8. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86 (2002)
9. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics, 313–330 (1993)