DETECTING SPAM REVIEWS ON SOCIAL MEDIA USING NETWORK-BASED FRAMEWORK: NETSPAM

[1]P Kokila, [2]Mir Mustafa Ali, [3]Nikhil H M, [4]Pooja Deshpande, [5]Pratima Kulkarni

[1]Assistant Professor, [2]UG Student, [3]UG Student, [4]UG Student, [5]UG Student,

[1]Department of Information Science & Engineering,
[1]The Oxford College of Engineering, Bangalore, India

Abstract : A lot of people rely on content available on social media for making decisions. The possibility that anyone can post a review provides a golden opportunity for spammers to write spam reviews about products and services. Identifying these spammers and the spam content is a very important topic in field of research and although a considerable number of studies have been done recently, but so far, the methodologies put forth still barely detect spam reviews, and none of them show the importance of each extracted feature type. This propose a novel framework, named *NetSpam*, which utilizes spam features for modeling review datasets as heterogeneous information networks to map spam detection procedure into a classification problem in such networks. Using the importance of spam features help us to obtain better results in terms of different metrics experimented on real-world review datasets from Yelp and Amazon websites. The results show that *NetSpam* is better than the existing methods using the features like review-behavioral, user-behavioral, review-linguistic, user-linguistic.

Keywords: NetSpam, Social Network, Spammer, Spam Review, Fake Review, Heterogeneous Information Network

I. INTRODUCTION

Information propagation is considered as an important source for producers in their advertising campaigns as well as for customers in selecting products and services. In the past years, people rely a lot on the written reviews in their decision-making processes, and positive/negative reviews encouraging/discouraging them in their selection of products and services. In addition, written reviews also help service providers to enhance the quality of their products and services. These reviews thus have become an important factor in success of a business while positive reviews can bring benefits for a company, negative reviews can potentially impact credibility and cause economic losses. The fact that anyone with any identity can leave comments as review provides a tempting opportunity for spammers to write fake reviews designed to mislead users' opinion. These misleading reviews are then multiplied by the sharing function of social media and propagation over the web. The reviews written to change users' perception of how good a product or a service are considered as spam and are often written in exchange for money Despite this great deal of efforts, many aspects have been missed or remained unsolved. One of them is a classifier that can calculate feature weights that show each feature's level of importance in determining spam reviews.

Spam minded informal conversations on social media (e.g. Twitter) shed light into their educational experiences, opinions, feelings, and concerns about the learning process. Data from such un-instrumented environments can provide valuable knowledge to inform student learning. Analyzing such data, however, can be challenging. The complexity of spam minded' experiences reflected from social media content requires human interpretation. However, the growing scale of data demands automatic data analysis techniques. Here data mining algorithm based on Spam filter is implemented which contains several steps like Data Collection from twitter, Cleaning the data by removing stop words, removal of non-letter and punctuation marks, probability of the words for various categories is estimated. For all the tweets Accuracy, Precision, Recall, F1 measure, Micro Averaged & Macro Averaged values are computed for each category and also for the various users. Therefore, its concluded based on average how many spam's minded have various categories of problems as well as extend this to the problems faced by which user.

Social media sites such as Twitter provide great venues for spam minded to share joy and struggle, vent emotion and stress, and seek social support. On various social media sites, spam minded discuss and share their everyday encounters in an informal and casual manner. Spam minded' digital footprints provide vast amount of implicit knowledge and a whole new perspective for educational researchers and practitioners to understand spam minded' experiences outside the controlled classroom environment. This understanding can inform institutional decision-making on interventions for at-risk spam minded, improvement of education quality, and thus enhance student recruitment, retention, and success. The abundance of social media data provides opportunities to understand spam minded' experiences, but also raises methodological difficulties in making sense of social media data for educational purposes. Just imagine the sheer data volumes, the diversity of Internet slangs, the unpredictability of locations, and timing of spam minded posting on the web, as well as the complexity of spam minded'

experiences. Pure manual analysis cannot deal with the ever-growing scale of data, while pure automatic algorithms usually cannot capture in-depth meaning within the data.

There is huge amount of data available in Information Industry. This data is of no use until converted into useful information. Analyzing this huge amount of data and extracting useful information from it is necessary. The extraction of information is not the only process that need to perform; it also involves other processes such as Data Cleaning, Data Integration, Data Transformation, Data Mining, Pattern Evaluation and Data Presentation. Once all these processes are over, we are now position to use this information in many applications such as Fraud Detection, Market Analysis, Production Control, Science Exploration etc.

Data Mining is defined as extracting the information from the huge set of data. In other words we can say that data mining is mining the knowledge from data. This information can be used for any of the following applications:

- Market Analysis
- Fraud Detection
- Customer Retention
- Production Control
- Science Exploration

II. METHODOLOGY

2.1 Hash tag Submission

This module is responsible for taking input the hash tags and then save the hash tags in the format of (HashTagID, HashTag and ProductID)

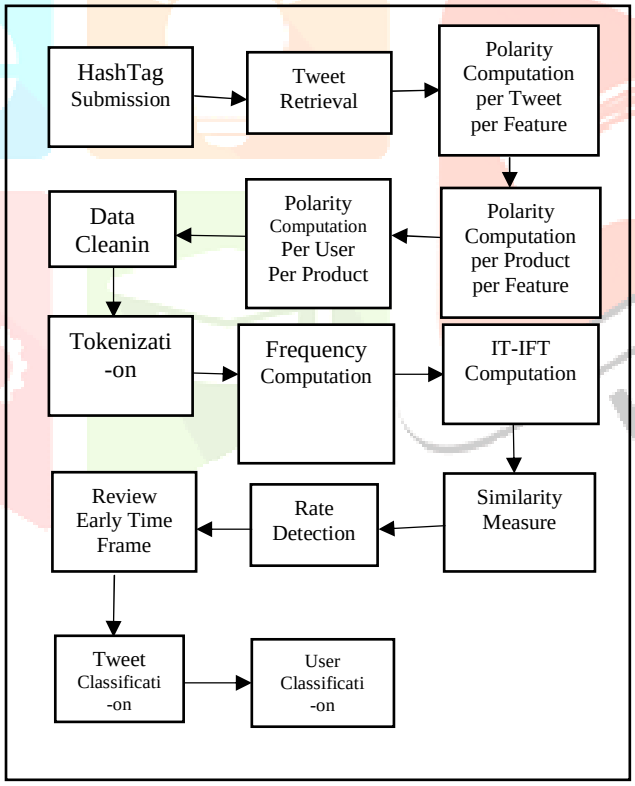


Fig.1: NetSpam Framework

2.1 Data Collection using Twitter

Twitter stores the reviews of the Products in the form of tweets which are associated with Hash Tags. This Module is responsible for Collecting tweets from Twitter by Passing the Hash Tag, APPID and Secret Key. APPID and Secret Key are unique generated IDs by twitter when application is created. Hash tag is a concept under which the users will be able to Tweet.

2.3 Polarity Computation per Tweet per Feature

This module is responsible for computing the sentiments of each tweet per feature. The positive sentiments, negative sentiments and neutral sentiments are found out per feature type. The feature types can be battery, memory, screen, touch and finally for each of the tweet the following matrix is computed.

Table.1: Tweet per feature

Tweet ID	Product ID	Feature Type	Positive Sentiment	Negative Sentiment	User ID
Unique ID for Tweet	Product ID for which tweet has been performed	It can be any feature type like- Battery, Memory,	Positive Sentiments for Tweet	Negative Sentiments for Tweet	Unique ID for the User

2.4 Polarity Computation per Tweet per Product

Polarity Computation per Tweet per Product is responsible for computation of polarity by computing the summation of polarities across tweets for the given product. Finally, the sentiment matrix can be defined as below

Table.2: Tweet per Product

Product ID	Feature Type	Positive Sentiment	Negative Sentiment	User ID
Product ID for which tweet has been performed	It can be any feature type like- Battery, Memory,	Positive Sentiments for Tweet	Negative Sentiments for Tweet	Unique ID for the User

2.5 User Based Sentiments

The set of unique users are found out and then for each of the user the sentiments are added upper product

Table.3: User Based Sentiments

Product ID	Positive Sentiment	Negative Sentiment	User ID
Product ID for which tweet has been performed	Positive Sentiments for Tweet	Negative Sentiments for Tweet	Unique ID for the User

2.6 Data Cleaning

Data Cleaning is used for removing the stop words from each of the tweets and clean them. After the data cleaning process is completed the clean data can be represented as a set CleanId ,CleanData ,UserId. CleanId is the unique Id associated with the Tweet, CleanData is the clean data after removal of clean data and UserId is the unique Id associated with the user.

2.7 Tokenization

The process of converting the statements into a sequence of words is called as tokenization

2.8 Frequency Computation

Frequency computation is a process of removing the repetition of tokens and hence removing the redundancy in the application. It is defined as number of times a token appears in the tweet

2.9 TF-IDF Computation

This is used to compute the inverse document frequency of each of the token and then multiply it by the text frequency.

$$IDF = log\ (N/f)$$

Where,

N = number of tweets in which tweet exist

f = frequency of word

The TF-IDF is computed using the following equation

$$TF - IDF = f*IDF$$

2.10 Similarity Measure

Similarity Measure is responsible for finding the unique tokens between the tweets and then finding whether the tweets are similar based on the number of intersections and number of unions. Ratio of intersection sum and union sum will give the similarity measure.

2.11 Rate Deviation

Difference between the reviews of each of the users if certain users have more of such difference those are regarded as spam.

2.12 Early Time Frame Measure

This module takes the tweets and measures the duration in which tweets are performed by the users and if there are any tweets which have been given within certain duration repeatedly negative for a product.

2.13 Classification of Tweet

It measures the weight by computing the similarity between the tweets and then finding the sentiments score and then find the weight. If the weight exceeds the certain threshold the tweet is classified as spam otherwise it is not classified as spam.

2.14 Classification of Spam User

This is responsible for finding whether the user is spam users or not based on user's-based sentiments and the similarity measure of user's-based tweets.

2.15 Metapath Definition and Creation

A metapath is defined by a sequence of relations in the network schema. Table.2 shows all the metapath used in the proposed framework. As shown, the length of user-based metapath is 4 and the length of review based metapath is 2.

For metapath creation, we define an extended version of the metapath concept considering different levels of spam certainty. In particular, two reviews are connected to each other if they share same value. Hassanzadeh*et al.* propose a fuzzy-based framework and indicate for spam detection, it is better to use fuzzy logic for determining a review's label as a spam or non-spam. Indeed, there are different levels of spam certainty. We use a step function to determine these levels. In particular, given a review u, the levels of spam certainty for metapathp_l (i.e., feature l) is calculated as$m_u^{p_l} = \frac{\lfloor s \times f(x_{l,u}) \rfloor}{s}$, where s denotes the number of levels. After computing $m^{p}{}_{u}{}^{l}$for all reviews and metapaths, two reviews u and v with the same metapath values (i.e., $m_u^{p_l} = m_v^{p_l}$) for metapathp_l are connected to each other through that metapath and create one link of review network. The metapath value between them denoted as $mpu,vl= mpul$.

Using s with a higher value will increase the number of each feature's metapaths and hence fewer reviews would be connected to each other through these features. Conversely, using lower value for s leads us to have bipolar values (which mean reviews take value 0 or 1). Since we need enough spam and non-spam reviews for each step, with fewer numbers of reviews connected to each other for every step, the spam probability of reviews take uniform distribution, but with lower value of s we have enough reviews to calculate final spamicity for each review. Therefore, accuracy for lower levels of s decreases because of the bipolar problem and it decades for higher values of s, because they take uniform distribution. In the proposed framework, we considered s = 20, i.e. $m^{p}{}_{u}{}^{l} \in \{0,0.05,0.10,...,0.85,0.90,0.95\}$.

Table.4: Features for users and reviews in four defined categories

Spam Feature	User Based	Review Based
Behavioral Based Features	*Burstiness* [20]: Spammers, usually write their spam reviews in short period of time for two reasons: first, because they want to impact readers and other users, and second because they are temporal users, they have to write as much as reviews they can in short time. $x_{BST}(i) = \begin{cases} 0 & (L_i - F_i) \notin (0,\tau) \\ 1 - \frac{L_i - F_i}{\tau} & (L_i - F_i) \in (0,\tau) \end{cases}$ (1) where $L_i - F_i$ describes days between last and first review for τ = 28. Users with calculated value greater than 0.5 take value 1 and others take 0. *Negative Ratio* [20]: Spammers tend to write reviews which defame businesses which are competitor with the ones they have contract with, this can be done with destructive reviews, or with rating those businesses with low scores. Hence, ratio of their scores tends to be low. Users with average rate equal to 2 or 1 take 1 and others take 0.	*Early Time Frame* [16]: Spammers try to write their reviews asap, in order to keep their review in the top reviews which other users visit them sooner. $x_{ETF}(i) = \begin{cases} 0 & (T_i - F_i) \notin (0,\delta) \\ 1 - \frac{T_i - F_i}{\delta} & (T_i - F_i) \in (0,\delta) \end{cases}$ (2 where $L_i - F_i$ denotes days specified written review and first written review for a specific business. We have also δ = 7. Users with calculated value greater than 0.5 takes value 1 and others take 0. *Rate Deviation using threshold* [16]: Spammers, also tend to promote businesses they have contract with, so they rate these businesses with high scores. In result, there is high diversity in their given scores to different businesses which is the reason they have high variance and deviation. $x_{DEV}(i) = \begin{cases} 0 & otherwise \\ 1 - \frac{r_{ij} - avg_{e \in E_{ij}} r(e)}{4} > \beta_1 \end{cases}$ (3 where β_1 is some threshold determined by recursive minimal entropy partitioning. Reviews are close to each other based on their calculated value, take same values (in [0,1)).
Linguistic Based Features	*Average Content Similarity* [7], *Maximum Content Similarity* [16]: Spammers, often write their reviews with same template and they prefer not to waste their time to write an original review. In result, they have similar reviews. Users have close calculated values take same values (in [0,1)).	*Number of first Person Pronouns, Ratio of Exclamation Sentences containing '!'* [6]: First, studies show that spammers use second personal pronouns much more than first personal pronouns. In addition, spammers put '!' in their sentences as much as they can to increase impression on users and highlight their reviews among other ones. Reviews are close to each other based on their calculated value, take same values (in [0,1)).

2.16 Classification

The classification part of *NetSpam*includes two steps; (i) *weight calculation* which determines the importance of each spam feature in spotting spam reviews, (ii) *Labeling* which calculates the final probability of each review being spam. Next we describe them in detail.

1) Weight Calculation: This step computes the weight of each metapath. We assume that nodes' classification is done based on their relations to other nodes in the review network; linked nodes may have a high probability of taking the same labels. The relations in a heterogeneous information network not only include the direct link but also the path that can be measured by using the metapath concept. Therefore, we need to utilize the metapaths defined in the previous step, which representheterogeneous relations among nodes. Moreover, this step will be able to compute the weight of each relation path (i.e., the importance of the metapath), which will be used in the next step (Labeling) to estimate the label of each unlabeled review.

The weights of the metapaths will answer an important question; which metapath (i.e., spam feature) is better at ranking spam reviews? Moreover, the weights help us to understand the formation mechanism of a spam review. In addition, since some of these spam features may incur considerable computational costs (for example, computing linguistic-based features through *NLP* methods in a large review dataset), choosing the more valuable features in the spam detection procedure leads to better performance whenever the computation cost is an issue.

To compute the weight of metapathp_i, for i= 1,...,L where L is the number of metapaths, we propose following equation:

$$W_{p_i} = \frac{\sum_{r=1}^{n} \sum_{s=1}^{n} mp_{r,s}^{p_i} \times y_r \times y_s}{\sum_{r=1}^{n} \sum_{s=1}^{n} mp_{r,s}^{p_i}}$$

where n denotes the number of reviews and $mp^{p}_{r,s}{}^{i}$is a metapath value between reviews r and s if there is a path between them through metapathp_i, otherwise $mp^{p}_{r,s}{}^{i}$= 0. Moreover, $y_r(y_s)$ is 1 if review $r(s)$ is labeled as spam in the pre-labeled reviews, otherwise 0.

2) Labeling: Let $Pr_{u,v}$be the probability of unlabeled review u being spam by considering its relationship with spam review v. To estimate Pr_u, the probability of unlabeled review u being spam, we propose the following equations:

$$Pr_u = avg(Pr_{u,1}, Pr_{u,2}, \ldots, Pr_{u,n})$$

where n denotes number of reviews connected to review u.

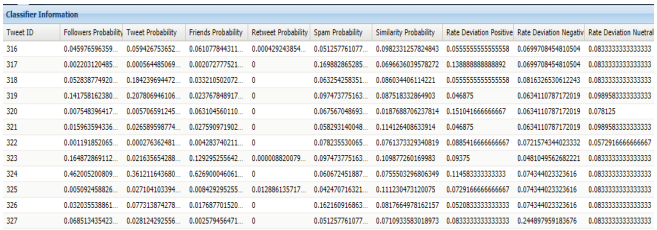
It is worth to note that in creating the HIN, as much as the number of links between a review and other reviews increase,its probability to have a label similar to them increase too, because it assumes that a node relation to other nodes show their similarity. In particular, more links between a node and other non-spam reviews, more probability for a review to be non-spam and vice versa. In other words, if a review has lots of links with non-spameviews, it means that it shares features with other reviews with low spamicity and hence its probability to be a non-spam review increases.

Table.5: Metapaths used in the NetSpam framework

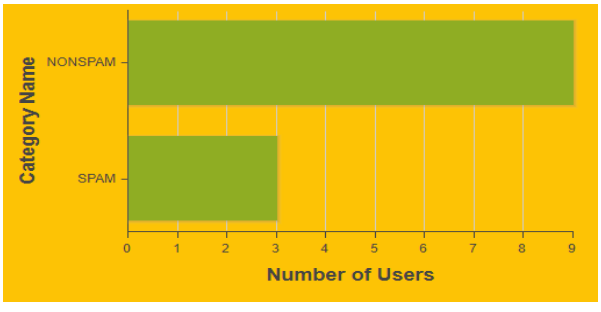
Row	Notation	Type	MetaPath	Semantic
1	R-DEV-R	RB	Review-Threshold Rate Deviation-Review	Reviews with same Rate Deviation from average Item rate (based on recursive minimal entropy partitioning)
2	R-U-NR-U-R	UB	Review-User-Negative Ratio-User-Review	Reviews written by different Users with same Negative Ratio
3	R-ETF-R	RB	Review-Early Time Frame-Review	Reviews with same released date related to Item
4	R-U-BST-U-R	UB	Review-User-Burstiness User-Review	Reviews written by different users in same Burst
5	R-RES-R	RL	Review-Ratio of Exclamation Sentences containing '!'-Review	Reviews with same number of Exclamation Sentences containing '!'
6	R-PP1-R	RL	Review-first Person Pronouns-Review	Reviews with same number of first Person Pronouns
7	R-U-ACS-U-R	UL	Review-User-Average Content Similarity-User-Review	Reviews written by different Users with same Average Content Similarity using cosine similarity score
8	R-U-MCS-U-R	UL	Review-User-Maximum Content Similarity-User-Review	Reviews written by different Users with same Maximum Content Similarity using cosine similarity score

III. CONCLUSION

This paper introduces a spam detection framework namely *NetSpam* based on a metapath concept as well as a new graph-based method to label reviews relying on a rank-based labeling approach. The performance of the proposed framework is evaluated by using two real-world labeled datasets of Yelp and Amazon websites. The observations shows that calculated weights by using this metapath concept can be very effective in identifying spam reviews and leads to a better performance. In addition, it is found that even without a train set, *NetSpam* can calculate the importance of each feature and it yields better performance in the features' addition process, and performs better than previous works, with only a small number of features. Moreover, after defining four main categories for features our observations show that the reviews behavioral category performs better than other categories, in terms of AP, AUC as well as in the calculated weights. The results also confirm that using different supervisions, similar to the semi-supervised method, have no noticeable effect on determining most of the weighted features, just as in different datasets.


Classifier Information

Tweet ID	Followers Probability	Tweet Probability	Friends Probability	Retweet Probability	Spam Probability	Similarity Probability	Rate Deviation Positive	Rate Deviation Negativ	Rate Deviation Nuetral
316	0.045976596359...	0.059426753652...	0.061077844311...	0.000429243854...	0.051257761077...	0.098233125782484 3	0.055555555555558	0.069970845481050 4	0.083333333333333
317	0.002203120485...	0.000564485069...	0.002072777521...	0	0.169882865285...	0.069663603957827 2	0.138888888888892	0.069970845481050 4	0.083333333333333
318	0.052838774920...	0.184239694472...	0.033210502072...	0	0.063254258351...	0.086034406114221	0.055555555555558	0.081632653061224 3	0.083333333333333
319	0.141758162380...	0.207806946106...	0.023767848917...	0	0.097473775163...	0.087518332864903	0.046875	0.063411078717201 9	0.098958333333333
320	0.007548396417...	0.005706591245...	0.063104560110...	0	0.067567048693...	0.018768870623781 4	0.151041666666667	0.063411078717201 9	0.078125
321	0.015963594336...	0.026589598774...	0.027590971902...	0	0.058293140048...	0.114126408633914	0.046875	0.063411078717201 9	0.098958333333333
322	0.001191852065...	0.000276362481...	0.004283740211...	0	0.078235530065...	0.076137332934081 9	0.088541666666667	0.072157434023332	0.057291666666667
323	0.164872869112...	0.021635654288...	0.129295255642...	0.000008820079...	0.097473775163...	0.109877260169983	0.09375	0.048104956268221	0.083333333333333
324	0.462005200809...	0.361211643680...	0.626900046061...	0	0.060672451887...	0.075553029680634 9	0.114583333333333	0.074344023323616	0.083333333333333
325	0.005092458826...	0.027104103394...	0.008429295255...	0.012886135717...	0.042470716321...	0.111230473120075	0.072916666666667	0.074344023323616	0.083333333333333
326	0.032035538861...	0.077313874278...	0.017687701520...	0	0.162160916863...	0.081766497816215 7	0.052083333333333	0.074344023323616	0.083333333333333
327	0.068513435423...	0.028124292556...	0.002579456471...	0	0.051257761077...	0.071093358301897 3	0.083333333333333	0.244897959183676	0.083333333333333



REFERENCES

1. S. Mukherjee, S. Dutta, and G. Weikum. Credible Review Detection with Limited Information using Consistency Features, In book: Machine Learning and Knowledge Discovery in Databases, 2016.
2. M. Luca and G. Zervas. Fake It Till You Make It: Reputation, Competition, and Yelp Review Fraud., SSRN Electronic Journal, 2016.
3. A. j. Minnich, N. Chavoshi, A. Mueen, S. Luan, and M. Faloutsos. Trueview: Harnessing the power of multiple review sites. In ACM WWW, 2015.
4. R. Shebuti and L. Akoglu. Collective opinion spam detection: bridging review networksand metadata. In ACM KDD, 2015.
5. Ch. Xu and J. Zhang. Combating product review spam campaigns via multiple heterogeneous pairwise features. In SIAM International Conference on Data Mining, 2014.
6. H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao. Spotting fake reviews via collective PU learning. In ICDM, 2014.
7. B. Viswanath, M. Ahmad Bashir, M. Crovella, S. Guah, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Towards detecting anomalous user behavior in online social networks. In USENIX, 2014G. Fei, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting burstiness in reviews for review spammer detection. In ICWSM, 2013.
8. L. Akoglu, R. Chandy, and C. Faloutsos. Opinion fraud detection in online reviews bynetwork effects. In ICWSM, 2013.
9. A. Mukerjee, V. Venkataraman, B. Liu, and N. Glance. What Yelp Fake Review Filter Might Be Doing?, In ICWSM, 2013.
10. A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh. Spotting opinion spammers using behavioral footprints. In ACM KDD, 2013.
11. M. Ott, C. Cardie, and J. T. Hancock. Estimating the prevalence of deception in online review communities. In ACM WWW, 2012.
12. S. Feng, R. Banerjee and Y. Choi. Syntactic stylometry for deception detection. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers; ACL, 2012.
13. N. Jindal, B. Liu, and E.-P. Lim. Finding unusual review patterns using unexpected rules. In ACM CIKM, 2012.
14. S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via temporal pattern discovery. In ACM KDD, 2012.
15. Y. Sun and J. Han. Mining Heterogeneous Information Networks; Principles and Methodologies, In ICCCE, 2012.
16. S. Feng, L. Xing, A. Gogar, and Y. Choi. Distributional footprints of deceptive product reviews. In ICWSM, 2012.
17. M. Ott, Y. Choi, C. Cardie, and J. T. Hancock. Finding deceptive opinion spam by any stretch of the imagination.In ACL, 2011.
18. G. Wang, S. Xie, B. Liu, and P. S. Yu. Review graph based online store review spammer detection. IEEE ICDM, 2011.
19. F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. Proceedings of the 22nd International Joint Conference on Artificial Intelligence; IJCAI, 2011.
20. Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In VLDB, 2011.
21. E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw. Detecting product review spammers using rating behaviors. In ACM CIKM, 2010.