Expert Systems With Applications 112 (2018) 148–155

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A unified framework for detecting author spamicity by modeling review deviation

Yuanchao Liu*, Bo Pang

School of computer science and technology, Harbin institute of technology, Harbin, China

ARTICLE INFO

Article history:
Received 16 December 2017
Revised 7 April 2018
Accepted 10 June 2018
Available online 14 June 2018

Keywords:
Review spam
Spam detection techniques
Fake reviews
Bidirectional LSTM
Review deviation

ABSTRACT

The success of e-commerce firms is highly dependent on the increasing number of customer reviews. However, to gain profit or fame, people may try to challenge the system by writing deceptive reviews that unjustly promote and/or demote target products or services. In this paper, a unified unsupervised framework is proposed to address the problem of opinion spamming. The rationale is that although not all outlier reviews are spam, spammers usually exhibit abnormities and deviations from normal users on certain dimensions concerning the same or even many products, thereby increasing their corresponding degrees of spamming (called "spamicity" in this paper). We introduce a set of abnormity signals from a review deviation angle and also present in detail an aspect-based review deviation dimension to model latent content deviation. Afterwards, a joint review deviation divergence is computed and ranked for detecting final opinion reviewer spamicity. Results of experiments conducted on a real-life Amazon review dataset demonstrate the effectiveness of the proposed approach.

1. Introduction

Product reviews are playing an increasingly important role in influencing the purchase decisions of potential customers. Most people prefer to browse or buy products with high reputation. According to a survey conducted by the US' Cone Communications,[1] 64% of customers refer to existing author comments before purchasing certain items. A Harvard study (Luca, 2011) also shows that 1% rise in star-rating increases revenue by 5%–9%. Thus, commodity producers or merchants pay much attention to review analysis to carry out economic activities, such as product recommendations, business strategy adjustment, and so on. Due to the financial incentives involved, spammers may be hired to write fake reviews to promote or demote the reputation of a product or service. Sellers may solicit biased reviews by incentivizing buyers to write reviews which does not appropriately reflect the quality of the product. True reviews are helpful for customers looking to buy the right product and for businesses aiming to improve product quality, whereas fake comments constitute interference or noise, and may have adverse effects on the Internet economy.

One of the approaches for opinion spam detection is supervised classification, which is mainly based on text content. A fake review often starts by praising certain products, belittling others, and inevitably exhibiting a number of external features (Ren, Ji, & Zhang, 2014). Therefore, researchers studied such fake reviews by applying certain models, such as NB (Naive Bayes), SVM (Support Vector Machine), and ME (Maximum Entropy), to train the classifier and then predict the unknown reviews (Heydari, Tavakoli, Salim, & Heydari, 2015; Feng, Banerjee, & Choi, 2012; Ott, Cardie, & Hancock, 2013; Ott, Choi, Cardie, & Hancock, 2011). Research on text classification is relatively mature and has been proven effective in other problems, such as e-mail spam and blog classification. Ott et al. (2011) used Amazon Mechanical Turk (AMT) to crowdsource anonymous online workers (called Turkers) who can write fake hotel reviews; they reported an accuracy of 89.6% using only certain word bigram features. Feng et al. (2012) boosted the accuracy to 91.2% using deep syntax features. However, the review spam corpus used in training and evaluation is mainly generated by AMT. Thus, the dataset is relatively small-scale and simulating complex fake reviews is difficult in actual scenarios.

By comparison, many unsupervised spam detection methods are based on behavior analysis (Lim, Nguyen, Jindal, & Liu, 2010; Mukherjee et al., 2013; Wang, Xie, Liu, & Yu, 2011). For example, businesses tend to hire certain campaign staff to submit fake reviews, which usually break out after a certain period of time, thus review burstiness can be exploited to identify the review spammer (Fei et al., 2013). In addition, reviewers may leave various behavioral footprints that can be utilized to facilitate the modeling of author spamicity (Mukherjee et al., 2013).

* Corresponding author.
E-mail addresses: ycliu@hit.edu.cn (Y. Liu), pangb@hit.edu.cn (B. Pang).

[1] http://www.conecomm.com/contentmgr/showdetails.php/id/4008.

https://doi.org/10.1016/j.eswa.2018.06.028
0957-4174/

Spam reviews usually look perfectly normal until one compares them with other reviews of the same product and identifies something inconsistent with the latter (Lim et al., 2010). A distributional divergence between two clusters, namely, spammers and non-spammers (Mukherjee et al., 2013), is apparent as opinion spammers may have different behavioral distributions from non-spammers. In the current paper, we propose a review deviation-based model (RDM) for spammer detection. The rationale is that although not all outlier reviews are considered spam, spammers usually exhibit abnormities and deviations from normal users on many dimensions and on many products, thus resulting in the corresponding increase in their spamicity. We introduce a set of abnormity signals from a review deviation angle and discuss in detail an aspect-based review deviation dimension to model latent content deviation. Then, the opinion reviewer spamicity detection is formulated as a ranking problem based on comprehensive distribution divergence. This work proposes a novel angle to solving the problem by modeling spamicity deviation, and its main contributions are listed below.

- We propose a novel review deviation model (RDM) for spammer detection.
- We introduce a set of abnormity signals from a review deviation angle, and provide details on the aspect-based review deviation dimension in order to model latent content deviation.
- We report empirical results on the Amazon review dataset, and the results demonstrate that our proposed approach is appropriate for this task.

The remainder of the paper is organized into sections. Section 2 discusses related works on opinion spam detection. Section 3 presents the framework of our proposed reviewer spamicity computation. Section 4 introduces a set of abnormity deviation signals, and details an aspect-based review deviation dimension to model latent content deviation. Afterwards, Section 5 empirically verifies the effectiveness of our method. Finally, Section 6 concludes the paper.

2. Related works

Deceptive opinion spam was first studied by Jindal and Liu (2008). We summarize related work as two kinds of approaches: supervised and unsupervised methods. The former is mainly based on text content, whereas the latter on author review behavior. In the last paragraph, we further detail the related works on review deviation.

Several related works regard fake review detection as a text classification problem. The results of such approaches are encouraging as very high accuracy has been achieved using only linguistic features. Li, Ott, and Cardie (2014) capture the general difference of language between deceptive and truthful reviews and argue that linguistic features may offer robust cues for spam review identification, such as highlighted sentiment or overuse of first person singular pronouns. Ren and Zhang (2016) use a neural network model to study the document-level representation for detecting deceptive opinion spam. Experimental results on three domains (Hotel, Restaurant, and Doctor) show that their method outperforms state-of-the-art methods in identifying deceptive opinion spam. Li, Ott, and Cardie (2013) introduce a semi-supervised manifold ranking algorithm for this task, which relies on a small set of labeled individual reviews for training. Experiments on a novel dataset of hotel reviews show that the proposed method outperforms many baselines. To leverage unlabeled data, Li, Huang, Yang, and Zhu (2011) proposed a semi-supervised co-training method, which is a typical bootstrapping method. The method starts with a set of labeled data, and increases the amount of annotated data by adding unlabeled data incrementally to boost performance. Hai et al. (2016) developed a multi-task learning method based on logistic regression, which can boost learning for a task by sharing the knowledge contained in the training signals of other related tasks.

The challenge for the existing supervised classification algorithms is that manually recognizing massive ground truth spam reviews is difficult given that labeling review spam is a labor-intensive task (Hai et al., 2016). Existing manually-labeled fake review datasets are relatively small in size. Xu, Shi, and Tian et al. (2015) also argue that the spamming degree of a review is a dyad that involves mutual interactions between the reviewer (user) and offering (item). Many features reveal the spamicity of a user. For example, Fei et al. (2013) model reviewers and their co-occurrence in bursts as a Markov random field, and employ the loopy belief propagation to infer whether a reviewer is a spammer or not in the graph. Savage, Zhang, and Chou (2015) and Lim et al. (2010) find that the reviews of opinion spammers tend to give extreme evaluation scores. Xu, Shi et al. (2015) propose a unified probabilistic graphical model to detect suspicious review spam, review spammers, and manipulated offerings. They additionally consider the abnormal features and text generality of an item to detect spamming. Wang, Liu, and He (2016) employ tensor decomposition to learn the embeddings of the reviewers and products in a vector space. They concatenate the review text, embeddings of the reviewer, and reviewed product as the representation of a review.

Detecting unfair ratings has been studied in several works (Dellarocas, 2000; Wu, Greene, & Smyth, 2010). Spammers are likely to deviate from the general rating consensus as spamming refers to incorrect projection. On a 5-star scale, deviation can range from 0 to 4 (Mukherjee et al., 2013). Mukherjee et al. (2013) estimate the posterior of this behavior. Their results show that only a small part (≈20%) of spammers have a deviation less than 2.5 and that most spammers have greater deviation compared to the rest, whereas most non-spammers (≈70%) are bounded by an absolute deviation of 0.6 (showing rating consensus). Fei et al. (2013) also argue that a reasonable reviewer is expected to give ratings similar to other reviewers of the same product. As spammers attempt to promote or demote products, their ratings can be quite different from other reviewers. Thus, rating deviation is a possible behavior demonstrated by a spammer. Lim et al. (2010) propose several scoring methods to measure the deviation degree of spam for each reviewer and apply them on an Amazon review dataset. Their results indicate that their ranking methods are effective in discovering spammers and can outperform other baseline methods based on helpfulness votes alone. Xu, Shi et al. (2015) also use the number of extreme ratings as one feature for their work. According to Feng et al. (2012), manipulated items that hire spammers to write spam reviews would necessarily distort the natural rating score distribution. Normally, the items involving spamming tend to have the bimodel (J-shaped) rating score distribution. The number of extreme rating (1 or 5 stars) is more than the number of 2 to 4 stars. Xu, Zhang, and Chang (2013) also measure how early ratings of a product deviate from the average rating of that product, concluding that early deviation may also mean spamicity.

3. Spamicity detection

In this work, we consider the problem of opinion spammer detection as a review deviation computation and ranking task. Fig. 1 summarizes the framework. Notably, we are not the first to exploit the concept of deviation for detecting spammer deviation. For example, Mukherjee et al. (2013) exploit the absolute rating deviation of a review from that of other reviews on the same product to measure the deviation of reviewers. They argue that more than 50% of the reviewers are unlikely to be spammers. By comparison,

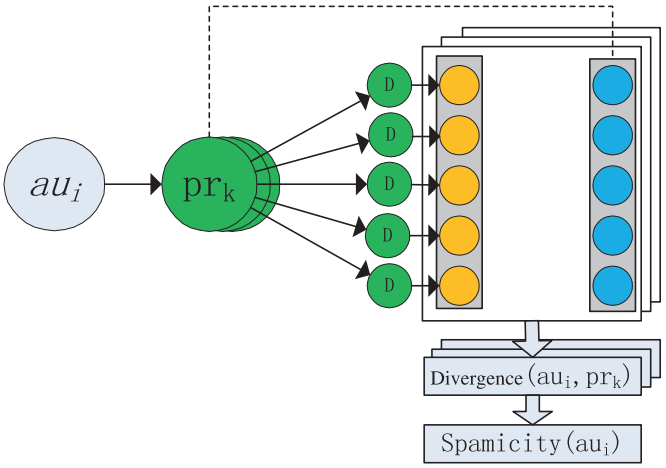

Fig. 1. The framework collectively utilizes author (au_i) metadata (review text, timestamp, rating), and review network (author metadata on the same product pr_k) under a unified framework to rank all users by spamicity which is computed from many spamicity dimensions.

Table 1
One sample review (ReviewerID: ID of the reviewer; Asin: ID of the product; ReviewerName: name of the reviewer; Helpfulness ratio: helpfulness ratio of the review(e.g., 40 of 50 people think the review is helpful); ReviewText: text of the review; Overall: rating of the product; Summary: summary of the review; ReviewTime: time of the review).

Name	value
ReviewerID	A3EXWV8FNSSFL6
Asin	B00DMWV3EU
ReviewerName	Daniel G. Lebryk
Helpfulness ratio	[40, 50]
ReviewText	The Anker Astro2 battery is a decent battery with a nice form factor and a soft exterior. The battery capacity (9000 mAh) is on par with others of this size. It includes an interesting accessory I like a lot.In the world of battery packs, it is the little things that separate these devices. ***
Overall	4.0
Summary	Good Battery - Cool Micro USB to 30 Pin Apple Adapter
ReviewTime	08 27, 2013

we define many deviation dimensions, as shown in Section 4. All these dimension feature vectors are computed to represent author spamicity, and the corresponding expectation vectors are also constructed using the relevant information posted by other authors.

One reviewer may review several products. The more product a reviewer writes about, the more behavior that reviewer exhibits; hence, greater spamicity may be detected with much confidence in this case. Li et al. (2011)) argue that spammers consistently write review spam, and their experiments show that most reviews (estimated about 85% in their paper) written by spammers are fake. In this paper, we mainly focus on spammer detection for users who write many reviews.[2]

We use the features in Section 4 to compute deviation. We set the expectation value of content duplication dimension for review-level feature to zero, then a bigger value means bigger spamicity if the review size is large enough. For other review-level dimensions (e.g., review length, rating score, aspect spamicity score, and helpful ratio), we use the variance of corresponding dimensions for all reviews on the current review product as the normal expectation value because there may be normal variances for genuine users. Review-level deviations are computed only if there are many other reviews for the product being talked about in the current review. For the author-level features, we use the average value of the corresponding dimensions for all authors as the dimension expectation.

Then *Spamicity*(*a*), i.e., the spamicity of author *a* can be calculated by using the equations

$$Spamicity(a) = DIV_{al}(a) + DIV_{rl}(a) \tag{1}$$

$$DIV_{al}(a) = JS(D_a||E_a)) \tag{2}$$

$$DIV_{rl}(a) = \underset{r \in r(a)}{avg}\,(JS(D_r||E_r)) \tag{3}$$

$$D_r = w_r * \{f_r\} \tag{4}$$

$$w_r = 1/\sqrt{rank_r}, \tag{5}$$

where $DIV_{al}(a)$ and DIV_{rl} mean divergence on the author- and review-level features, respectively. For review-level features, DIV_{rl} is calculated as the average of divergence of all reviews of one author. D_a and D_r represent each author's author- or review-level distribution, and E_a and E_r are the corresponding expectation distributions. We use the Jensen–Shannon divergence (JS) (Hinrich & Manning, 1999) to calculate the review deviation as JS is symmetric and satisfies the triangle inequality. JS is calculated using KL divergence (Kullback–Leibler divergence), which is commonly used to compare two distributions, to reflect how each author' review distribution deviates from the distribution of the expectation. In addition, w_r denotes early deviation factor, as Lim et al. (2010) verified in their work that early deviation captures the behavior of a spammer contributing a review spam immediately after a product is made available for review. Such spams are likely to attract attention from other reviewers, allowing spammers to manipulate the views of subsequent reviewers. Here $rank_r$ refers to the rank order of r among ratings assigned to the rated product. We set $rank_r = i$ for the ith rating (or review) of a product.

4. Spamicity behavior deviations

4.1. Behavior deviation

Certain behavior features can give a hint on the existence of spamicity (Mukherjee, Liu, & Glance, 2012; Wang et al., 2011; Rayana & Akoglu, 2015). From deviation based angle, we implemented many suspicious dimensions for review *r* written by author *a*. The value range of these behavior deviation features is normalized in [0, 1], where a bigger value usually means a bigger suspicious deviation. Although we cannot definitely say one is a spammer if spamicity is big for one dimension, the author can certainly become more suspicious if the spamicity values are bigger on several dimensions and if they exhibit such characteristics on a number of products. Some features may need additional information on other reviews. In Table 1, we give one sample review, which can also demonstrate the form of the input data. We use NLTK[3] to preprocess the review text, e.g., tokenization and POS tagging.

We will present these features from the review- and author-levels as follows. Note that for the review-level features, Section 4 also computes deviation distribution divergence on all

[2] This does not mean the reviewers who write only one review (singleton review) are non-spammers. For singleton review spam detection, please refer to Xie, Wang, and Lin (2012)).

[3] http://www.nltk.org/.

reviews of one author, which are found on many different products, to form a comprehensive spamicity for one author.

Review-level Features

- **Content duplication** $f_{dup}(r)$

Content duplication can be computed using

$$f_{dup}(r) = \begin{cases} \max\limits_{\{c|p(c)=p(r),T_c<T_r\}} (sim(c,r)), & if\ len(r) > N_{thr} \\ & and\ r\ is\ not\ the\ first\ review\ for\ p(r) \\ 0.5, & else \end{cases} \quad (6)$$

where $p(r)$ denotes the product item that r reviews about, T_r means the review time for r, $len(r)$ denotes the number of tokens in review r, $c$denotes the reviews which is about $p(r)$ and earlier than r. The intuition is that writing one's imagined experience is difficult. Hence, spammers may choose to copy other reviews. This formula computes the similarity between review text r and earlier reviews on the same product. We use Vector Space Model (Salton, 1975) plus cosine similarity to compute content similarity between c and r, i.e., $sim(c,r) = \cos(v_c, v_r) = \sum_{i=1}^{n}(v_c \times v_r)/(\sqrt{\sum_{i=1}^{n}(v_c)^2} \times \sqrt{\sum_{i=1}^{n}(v_r)^2})$, in which the vector space is constructed according to all the tokens in both c and r, $n$is the dimension number, v_c and v_r are the vector of c and the vector of r separately. If r contains only a few words (e.g., less than 10 words), or r is the first review for the product, we set $f_{dup}(r) = 0.5$ as making a distinction is difficult.

- **Review length** $f_{len}(r)$

Review length can be expressed as

$$f_{len}(r) = mms(|len(r) - avg_{c \in p(r)}(len(c))|), \quad (7)$$

$$mms_{p(x)}(x) = (x - min)/(max - min). \quad (8)$$

We use the Min–Max-Scaler to normalize the review length. Here, $len(r)$ denotes the word count in review r, $avg_{c \in p(r)}$ is the average review length for all reviews of the product item that current review r talks about, min and max are the minimum and maximum values of review length for all reviews of the product item that current review r talks about.

- **Rating score** $f_{rs}(r)$

The rating score can be obtained by using the equation

$$f_{rs}(r) = mms(|rs(r) - avg_{c \in p(r)}(rs(c))|), \quad (9)$$

where $rs(r)$ is the rating score for review r, and $avg_{c \in p(r)}(rs(c))$ is the average score for all reviews of the product item that the current review r talks about. We also use the Min–Max-Scaler to normalize the score deviation, which is similar to that formula (8).

- **Helpful ratio** $f_{hr}(r)$

The helpful ratio can be computed using

$$f_{hr}(a) = mms(|hr(r) - avg_{c \in p(r)}(hr(c))|), \quad (10)$$

where $hr(r)$ denotes the helpful ratio (percentage of people who think that the review is helpful) for one review r, and $avg_{c \in p(r)}(hr(c))$ is the average help ratio for all reviews of the product item that current review r talks about. We use this dimension because helpful scores may be overused (Mukherjee et al., 2013). Note that we do not assume that a small helpful score automatically means that a review is spam in this dimension.

Author-level Features

- **Reviewing burstiness**

Here, $f_{burst}(a)$ is given by

$$f_{burst}(a) = \begin{cases} 0, & if\ T_L(a) - T_F(a) > \tau_B \\ 1 - \frac{T_L(a) - T_F(a)}{\tau_B}, & otherwise \end{cases}, \quad (11)$$

where τ_B represents the time span threshold (we set it to 30 days in this paper), $T_F(a)$ and $T_L(a)$ represent the first and last review time of author a separately. This is based on the assumption that spammers usually post reviews in a short time span.

- **Extreme rating** $f_{ext}(a)$

This can be obtained using the expressions

$$f_{ext}(a) = avg_{r \in r(a)}|IsExt(r)|, \quad (12)$$

$$IsExt(r) = \begin{cases} 1, & if\ rs(r) \in \{\text{minRS}, \text{maxRS}\} \\ 0, & else \end{cases}, \quad (13)$$

where $IsExt(r)$ represents whether review r gives an extreme rating score. MinRS and maxRS denote minimal and maximal rank separately. On a 5-star (*) rating scale which is widely adopted by e-commerce websites, extreme ratings mean 1* or 5*. We leverage this behavior because spammers tend to give extreme rating scores in order to demote/promote product.

- **The purity of sentiment in review**

Here, $f_{pur}(a)$ is expressed as

$$f_{pur}(a) = avg_{r \in r(a)}|DPWords(r)|/|PWords(r)| \quad (14)$$

where $DPWords(r)$ represents sentiment words with dominant polarities (*pos.* or *neg.*) in review r, and $PWords(r)$ represents the sentiment words in r. For negation, we simply reverse the polarity of sentiment words where negation words are nearby (Pang, Lee, & Vaithyanathan, 2002). We use Liu Bing's Opinion Lexicon[4] (Hu & Liu, 2004) to estimate the sentiment purity and omit the smoothing factor. We leverage this behavior because spammers tend to give extreme sentiment and use words with one kind of polarity (e.g., *neg.*) much more than the opposite (e.g., *pos.*) in their reviews (Peng & Zhong, 2014; Elmurngi & Gherbi, 2017).

- **Maximum Number of Reviews**

This can be obtained using the expression

$$f_{mnr}(a) = \begin{cases} mms(max\_r(a)), & if\ max\_r(a) > 1 \\ 0, & else \end{cases} \quad (15)$$

where $max\_r(a)$ is the maximum number of reviews in a single day for author a. We also use the Min–Max-Scaler to normalize the dimension by the maximum value $max\_r(a)$ of all authors, which is similar to formula (8). Posting many reviews in a single day indicates an abnormal behavior.

- **Ratio of early reviews** $f_{rer}(a)_a$

Here, $f_{rer}(a)_a$ can be obtained using the equation

$$f_{rer}(a) = |\{r \in R_a : T_r \le T_F(p(r)) + \tau_F\}|/|R_a| \quad (16)$$

where R_a means all the reviews by author a, T_r represents the time at which review r can be obtained is posted, $T_F(p(r))$ represents the time at which the first review for product $p(r)$ is posted, and τ_F refers to the time span for the first reviews (30 days in this paper).

[4] https://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar.

4.2. Aspect opinion deviation

Several existing methods for opinion spam detection focus on behavior analysis (Fei et al., 2013; Mukherjee et al., 2013). However, mining for meaningful textual features tends to be ignored in the literature. Intuitively, like extremity or deviation in rating score, spammers prefer to express extreme opinions toward aspects in their reviews regardless of the truth as they deliberately write fake reviews, thus exhibiting more review abnormities or deviations. Although spammers may take certain anti-detection measures (e.g., writing with enough length or in more detail) to make their reviews look genuine, expressing the reasonable sentiments towards the product details is still difficult for them. They always tend to express extreme opinion in their reviews when most of them do not even have any purchase experience. Given the motive-driven nature of spamming activities, review spammer detection thus requires an approach from another dimension, such as review length and unhelpful review detection.

Then, we further exploit aspect-based opinion deviation and use it as another review-level deviation. For example, one spammer may hold a positive opinion about the "price" of one product by saying "*the price of this device is reasonable*" for certain product items, whereas most people hold a negative opinion toward this aspect. Naturally, one user may be different from others on several dimensions due to the variance; however, users who exhibit abnormities on many dimensions or products tend to increase their spamicity (Mukherjee et al., 2013; Wu et al., 2010). Mukherjee et al. (2013) argue that more than 50% of the reviewers are unlikely to be spammers, and Xu, Ba, and Kiros et al. (2015) report from 8% to 15% spam rates in their studies. By comparison, genuine reviewers usually describe opinions according to their own real experiences and show reasonable variance. It can also be seen from experiment results (see Section 5) that by adding aspect-based opinion deviation, the performance gains improvement over the behavior features in Section 4.1.

4.2.1. Aspect extraction

We first count the frequency of all words that have noun-characteristics (POS = "NN", "NNS", "NNP", and "NNPS", which are the labels of noun types. In detail, "NN" means singular noun, "NNS" means plural noun, "NNP" means singular proper noun, "NNPS" means plural proper noun.[5]) in the 10, 000 reviews[6] of each dataset used in this paper, as shown in Section 5, and sorted them by frequency. Most of them appear only several times and are not the aspects that can be considered suitable. Next, to guarantee that targets are valid and the aspects are associated with the items, we select the first 400 frequent nouns that have a frequency bigger than 40, after which we manually check whether these candidate words are suitable as aspects.[7] We also manually check some less disambiguated terms as the candidate aspect set. For example, when describing cell phone screen and a portable charger, the aspect "size" is associated with different sentiments. We believe that they are suitable for calculating aspect opinion deviation as focused on by most users.

Table 2
Aspect star tagging scheme (1 and 2 stars = Negative, 4 and 5 stars = Positive).

Review star	Aspect star	
	Pos.	Neg.
5	5	2
4	4	2
3	–	–
2	3	2
1	3	1

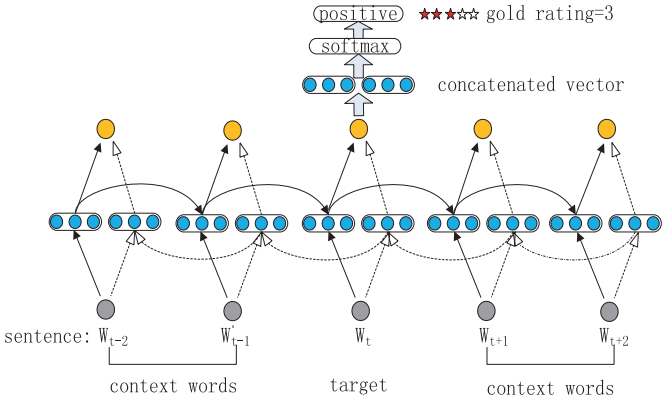

Fig. 2. Bidirectional LSTM for aspect opinion classification. W_t denotes the aspect word, and all words except W_t are context words. Sentiment is dependent not only on preceding words but also on successive ones. The concatenation of the corresponding states of the forward and backward LSTM is used for sentiment classification.

4.2.2. Aspect star tagging scheme

While writing reviews on many e-commerce websites, the authors usually give one overall rating score. This review–score relation is an important hint for user opinion on aspects in the review. For example, by writing a 1-star review, users show very negative sentiments toward these aspects, whereas in a 5-star review, users show positive sentiments toward aspects. By examining the dataset, we find certain exceptions in which the review-level star may be not suitable for all aspects of each sentence in it. We also note that in the same sentence, more than one aspect and opinions vary. For example, in a 5-star sentence "*graphics are appealing, but the game has to narrow of a scope to keep a person playing,*" two aspects can be found: "graphics" and "game." The former is positive, whereas the latter is negative. We find that for such cases, most of them have words such as "but," "however," "whereas" and so on. We then use these hint words to find the exceptions and reverse the opinions correspondingly. Then "graphics" is given a score of 5 (positive) referring to the review star and "game" is given 2 (negative). The aspect star tagging scheme is summarized in Table 2. Note that in Table 2, we only keep the 4–5 star (positive) and 1–2 (negative) star reviews, because knowing the aspect score when one user gives a middle overall score of 3 is difficult.

4.2.3. Aspect opinion

We mainly use Bi-LSTM (Bidirectional Long Short-Term Memory) (Graves, Mohamed, & Hinton, 2013) for aspect opinion classification. Sentiment is dependent not only on preceding words but also on successive ones. Bi-LSTM allows us to look ahead by employing a forward LSTM, which processes the sequence in chronological order, and a backward LSTM, which processes the sequence in reverse order. The output at a given time step is then the concatenation of the corresponding states of the forward and backward LSTM as shown in Fig. 2.

[5] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[6] Take the dataset Cell Phones and Accessories as one example, the top selected top words include: case: 3300; charger: 2852; product: *1886; battery: 1560; price: 1286; quality: 931*; and *headset: 908*. We also count frequency with different sizes and find that when size is bigger than 1,000, the top frequent 400 nouns are very steady. Implicit aspects in the review is also noted where the opinion is about hidden aspects. We did not consider such a case in this paper.

[7] Although it is natural that other existing opinion aspects extraction methods can be used instead, our concern is to guarantee the validation of targets as we found that even the state-of-art methods may also extract some invalid aspects. In fact, our practice shows that manual verification work is efficient (only cost several hours) as there are not so many widely used aspects for one kind of product.

To build the target sentiment classifier, we use softmax, which is calculated as

$$y(a) = softmax(W_s a + b_s), \tag{17}$$

$$softmax_i = \frac{exp(x_i)}{\sum_{j=1}^{C} exp(x_j)}, \tag{18}$$

where $W_s \in R^{5 \times d}$ is the sentiment classification matrix and transforms composition vector a to a real-valued vector, whose length is class number C (e.g., 5, as product reviews have 1–5 stars). Then, Softmax converts real values to conditional probabilities.

We use cross-entropy error between the gold sentiment distribution $t^i \in R^{C \times 1}$ and predicted sentiment distribution $y^i \in R^{C \times 1}$ as the loss function, which is expressed as

$$loss(\theta) = \sum_{i \in T} \sum_{j=1}^{C} t_j^i \bullet log(y_j^i) + \lambda ||\theta||^2, \tag{19}$$

where T is the training data. The loss is a function of parameters $\theta = (W, b, W_s, b_s, L)$. All word vectors are stacked in the word embedding matrix $L \in R^{d \times |V|}$, where $|V|$ is the size of the vocabulary. The word vectors are initialized with 200-dimensional word embeddings, which are learned with SkipGram (Mikolov, Sutskever, & Chen, 2013) before training. Other parameters in θ are initialized from a uniform distribution $\mu(-r, r)$, where $r = 0.01$. Thus, we set the learning rate at 0.03. Back propagation is employed to propagate the errors from the top to the leaf nodes. Derivatives of parameters are used to update the parameters. We remove sentences with no aspects and unroll the aspects of every sentence in the review (e.g., a sentence with two aspects occurs twice in succession, once with each aspect). Next, we train our model to minimize cross-entropy loss, using stochastic gradient descent, the Adam update rule (Kingma & Ba, 2015), mini-batches of size 10, and early stopping with a patience of 10.

Given the abovementioned aspect opinion, spamicity score can be computed by using the equation for aspect spamicity score $f_{ass}(r)$. This is given by

$$f_{ass}(r) = mms(ac(r)), \tag{20}$$

where $ac(r)$ is the aspect deviation count for all aspects in review r, and $avg_{c \in p(r)}(ac(c))$ is the average aspect deviation count for all reviews of the product item that current review r talks about. To better capture the deviation, 4- and 5-star reviews are regarded as positive and 1- and 2-star reviews are regarded as negative. Thus, for one aspect in review r, if the dominant opinions in all reviews are opposite, the aspect deviation count will add one. This review-level deviations are computed only if there are at least 5 other reviews on this aspect for the product about which the current review talks about. We also use the Min–Max-Scaler to normalize the score deviation, which is similar to formula (8).

5. Experiments

5.1. Dataset

We conduct experiments on four Amazon review datasets (Mcauley, Pandey, & Leskovec, 2015) for evaluation.[8] These datasets have a large review/product ratio, and may thus exhibit more clear behaviors. To better capture the deviation, we compute spamicity for authors who write at least 5 reviews. For each review, spamicity is computed when there are at least 30 reviews for the same product item. The datasets contains product reviews, review ratings, text, helpfulness votes, and product metadata. Finding all the reviews written by one reviewer or those about one product in the dataset is convenient for us. The details of these datasets are shown in Table 3. Below, we describe the baseline systems and evaluation results.

5.2. Results and analysis

5.2.1. Performance evaluation

For a comprehensive comparison, we also experiment with state-of-the-art unsupervised approaches. (1) Feature Sum (FSum) ranks the authors in descending order of the sum of all abnormal behavior feature values where deviation from the expectation is not incorporated. (2) Helpfulness Score (HS) assumes that spam reviews should receive less helpfulness feedback (Zhang et al., 2015). HS rank reviews the average helpfulness rating of the reviews posted by the same author in ascending order. Two popular learning to rank methods have been introduced in the literature: (3) SVMRank (Joachims, 2002) and (4) LambdaMART in Ranklib.[9] Each training ranking is produced by sorting authors by values of a spamicity feature. Each author is represented as a vector of 10 spam features given that there are 10 features in total (In order to change review-level features into author-level ones, we take the expected feature value of all reviews of each author). By using all 10 features, including aspect opinion deviation, we test two variants of our model: (5) RDM1, which is the RDM variant which use 9 features in Section 4.1, and (6) RDM2, which is the fully unsupervised version of our review deviation-based author spamicity detection model.

We adopt the review classification method (Mukherjee et al., 2013; Xu, Ba, et al., 2015) for evaluating the review spammer ranking results. The idea is that if the rank model is effective, the top k% reviewers will be prone to be spammers than the bottom k% reviewers. Then two different review datasets can be formed. Word usage difference between spammers and non-spammers has also been demonstrated. Supervised text classification using n-gram features is effective in detecting spam and non-spam reviews (Li et al., 2014; Mukherjee et al., 2013) . Then, the resulting training data can be fed into n-gram-based classification to evaluate the performance of the model. Notably, the textual n-gram features are not used in the model. This is binary classification problem for performance evaluation, so we use the default evaluation method, i.e., spam class is deemed as positive class for computing the precision, recall, and F1-measure.

Detecting review spam is a challenging task because no one knows exactly the amount of spam in existence (Lim et al., 2010). Deception prevalence studies (Mukherjee et al., 2013; Xu, Shi et al., 2015) report anywhere from 8% to 15% spam rates in their studies. Hence, to provide the ground-truth spam reviews and non-spam reviews for review classification, we use top 10% and bottom 10% as the spam reviewers and non-spam reviewers separately. We perform the classification using svm_light (Joachims, 1998) with a linear kernel. Table 4 reports evaluation results with 5-fold cross-validation.

From the results shown in Tables 4(a)–(d), it is promising to see that both of our review deviation-based author spamicity detection models RDM1 and RDM2 outperform all baseline methods over each dataset.

First, the performance of RDM1, which uses behavior deviation features, is better than other baselines. This indicates that

[8] Certain text-based datasets for review spam problem (e.g., annotated data from (Mukherjee et al., 2013) and the one proposed in (Li, Ott, & Cardie, 2014)) are also noted. These datasets contain only the true or fake review texts, which are written by hired Turkers and do not contain user or product information. They are not suitable for our task, because we focus on using users' deviation on many dimensions, for which we utilize relation between review, user, and product.

[9] http://people.cs.umass.edu/~vdang/ranklib.html.

Table 3
Statistics of data sets.

Data Set	#Review	#Author	#Item	Avg. words
Reviews Cell Phones and Accessories	731,842	68,010	17,437	61.62
Tools and Home Improvement	258,963	29,764	7,670	95.24
Toys and Games	330,417	34,786	8,367	89.46
Office Products	98,914	10,532	5,261	98.19

Table 4
Performance evaluation over different datasets.

Methods	P	R	F1	Accu.
(a) Cell phones and accessories				
HS	61.28	60.43	60.85	60.31
Fsum	65.34	64.03	64.68	64.78
SVMRank	66.42	66.62	66.52	66.32
LambdaMART	66.94	67.28	67.11	67.61
RDM1	70.02	68.24	69.12	69.32
RDM2	**71.35**	**70.83**	**71.09**	**71.18**
(b) Tools and home improvement				
HS	62.57	60.26	61.39	62.69
Fsum	67.32	68.21	67.76	68.08
SVMRank	67.79	68.84	68.31	67.32
LambdaMART	67.93	69.05	68.49	68.16
RDM1	68.06	70.31	69.17	68.24
RDM2	**75.21**	**74.63**	**74.92**	**73.91**
(c) Toys and games				
HS	67.13	71.84	69.41	68.72
Fsum	70.61	73.11	71.84	71.94
SVMRank	72.89	73.34	73.11	71.25
LambdaMART	73.91	73.69	73.80	72.19
RDM1	74.07	73.95	74.01	72.60
RDM2	**79.48**	**78.78**	**79.13**	**78.62**
(d) Office products				
HS	65.28	71.33	68.17	66.5
Fsum	68.77	72.17	70.43	67.97
SVMRank	70.25	73.49	71.83	68.52
LambdaMART	71.07	73.83	72.42	69.36
RDM1	72.52	74.21	73.36	70.27
RDM2	**78.67**	**79.49**	**79.08**	**78.15**

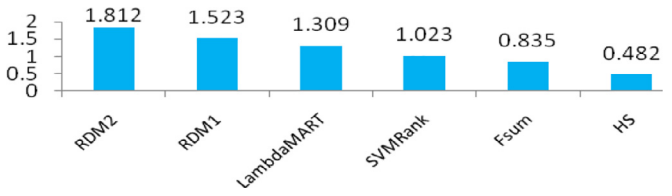

Fig. 3. Average distribution divergence between reviews of spammers and those of non-spammers (higher is better).

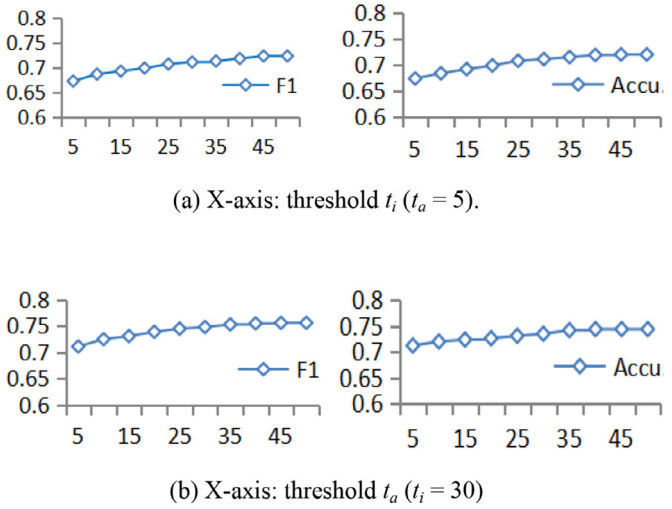

Fig. 4. Effect of t_a, and t_i on performance (k = 10).

our consideration for review deviation-based spamicity detection method is useful. Second, by adding aspect-based opinion deviation, the performance of RDM2 gains improvement in F1 and accuracy. This suggests that, by incorporating aspect-based review deviation into the existing behavior-based anomaly analysis, our model can complement for behavior features and contribute to the existing spammer review detection features. Although spammers may try to make their reviews look genuine, expressing true sentiment toward the product details is often difficult for them. They always tend to express extreme opinion in their reviews or do not even have any experience. Thus, their reviews usually exhibit aspect-level deviation.

That the performance of HS is poorer is noted, which suggests it is not suitable for opinion spammer detection. The reason may be two-fold: newly-posted reviews may get less helpful votes as they are reviewed by less readers; besides, helpful scores are subject to abuse. (Lim et al., 2010) argued that detecting spam and predicting helpfulness are two separate problems, because not all unhelpful reviews are spam. A poorly written review can be unhelpful but it may not be a spam. Except for HS, both RDM1 and RDM2 are better than FSUM. The reason may be that FSUM makes a mere sum of abnormal features and is not able to incorporate the deviation factor, whereas spammer behaviors are different from genuine authors on many dimensions. Furthermore, as each method uses different reviews for classification and output rankings from the systems are different, no significance test is applied here to compare the performances of different methods.

An extra advantage of leveraging deviation may be that these features are robust and hardly gamed by spammers. Consider the feature "Review Length" as an example, Mukherjee et al. (2013) find that the average review length of the spammers is quite short compared with that of non-spammers. However, once spammers realize this, they produce a review that is as lengthy as possible, and in such cases, the system finds it hard to make the distinction between spam and authentic content.

5.2.2. Analysis

In this subsection, we conducted further experiments and analyses to explain the difference between spammers and non-spammers in our approach. Taking the first dataset Cell Phones and Accessories as one example, we analyze the word distributions in different methods and plotted these in Fig. 3. The word distributions can explain why performances vary because text classification-based evaluation metrics relies on word distribution. We merge all spammers' reviews into one document and those of non-spammers' into another document and then compute word probability distribution difference among users using KL divergence. Fig. 3 shows that the RDM2 achieves the highest KL value, whereas HS has the least. Similar trends are also observed for other datasets.

As our approach is based on review deviation, we conjecture that when more reviews of one author or a product are used, more deviations may be discovered. We test this hypothesis by analyzing

the effect of the number of reviews for the same product item, or that of the number of reviews for same author, on opinion spammer detection performance. Again, over the dataset Cell Phones and Accessories, Fig. 4 depicts the metrics of F1 and accuracy by varying threshold t_a and t_i, where t_a means the authors who post at least t_a reviews are considered, and t_i means the items which have at least t_i reviews are considered.

In Fig. 4, we observe that performances consistently improve when a large review size threshold is considered. This improvement can be attributed to the fact that when more reviews for the same items are utilized, more review deviation features can be leveraged as referring benchmarks for spotting abnormal spammers due to the characteristics of our approach. Similarly, authors with more reviews exhibit more signals for review deviation detection. Similar trends are also observed for other datasets. However, they are omitted in this paper for brevity.

6. Conclusions and future works

In this paper, we propose a unified, unsupervised framework to address opinion spammer detection by modeling review deviation. The rationale is that spammers usually have no purchase- or utilization-experiences of the product. Most of them may have been hired to promote or demote the reputation of products or services. Consequently, their reviews would exhibit abnormities and deviations from normal users on many dimensions on the same or even different products. As it is very challenging to detect spammers, examining spamicity by using more dimensions is necessary. We defined several deviation dimensions to model reviewer spamicity. We also leverage target-dependent sentiment information to model content-based review deviation. As far as we know, this is the first trial based on deep content analysis. By comparison, our unified, unsupervised review deviation-based framework model can leverage many effective signals to detect spammers. Experimental results conducted on real-life Amazon review datasets demonstrate the effectiveness of our proposed approach. For future works, we plan to explore more methods of analysis for modeling the deviation. Additionally, further work can be pursued to explore semantically-based review deviations for review spam modeling.

Acknowledgments

We thank Julian McAuley from UCSD for his kind help on providing the dataset of this work. We thank our anonymous reviewers for helpful suggestions and corrections. This research was supported in part by the National Natural Science Foundation of China under contract 61672192 and 61572151.

References

Dellarocas, C. (2000). Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *ACM conference on electronic commerce* (pp. 150–157). ACM.

Elmurngi, E., & Gherbi, A. (2017). An empirical study on detecting fake reviews using machine learning techniques. In *International conference on innovative computing technology* (pp. 107–114).

Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R. (2013). Exploiting burstiness in reviews for review spammer detection. In *ICWSM* (pp. 175–184).

Feng, S., Banerjee, R., & Choi, Y. (2012). Syntactic stylometry for deception detection. In *Meeting of the association for computational linguistics: Short papers* (pp. 171–175). Association for Computational Linguistics.

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*(3), 6645–6649.

Hai, Z., Zhao, P., Cheng, P., Yang, P., Li, X. L., & Li, G. (2016). Deceptive review spam detection via exploiting task relatedness and unlabeled data. In *Conference on empirical methods in natural language processing* (pp. 1817–1826).

Heydari, A., Tavakoli, M. A., Salim, N., & Heydari, Z. (2015). Detection of review spam: A survey. *Expert Systems with Applications, 42*(7), 3634–3642.

Hinrich, S., & Manning, C. D. (1999). *Foundations of statistical natural language processing* (p. 304). Cambridge, Mass: MIT Press.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Tenth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 168–177). Washington, USA: Seattle.

Jindal, N., & Liu, B. (2008). Opinion spam and analysis. In *International conference on web search and data mining* (pp. 219–230). ACM.

Joachims, T. (1998). Making large-scale SVM learning practical. In *Making large-scale SVM learning practical* (pp. 499–526). MIT-Press.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 133–142).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR*: 2015.

Li, F., Huang, M., Yang, Y., & Zhu, X. (2011). Learning to identify review spam. In *Proceeding of the twenty-second international joint conference on artificial intelligence* (pp. 2488–2493).

Li, J., Ott, M., Cardie, C., et al. (2013). Identifying manipulated offerings on review portals. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1933–1942).

Li, J., Ott, M., Cardie, C., et al. (2014). Towards a general rule for identifying deceptive opinion spam. In *Meeting of the association for computational linguistics* (pp. 1566–1576).

Lim E, P., Nguyen V, A., Jindal, N., & Liu, B. (2010). Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 939–948).

Mcauley, J., Pandey, R., & Leskovec, J. (2015). Inferring networks of substitutable and complementary products. *Knowledge Discovery and Data Mining, 5*(4), 785–794.

Luca, Michael (2011). *Reviews, reputation, and revenue: The case of Yelp.com*. Harvard Business School Working Papers.

Mikolov, T., Sutskever, I., Chen, K., et al. (2013). Distributed representations of words and phrases and their compositionality. In *International conference on neural information processing systems* (pp. 3111–3119). Curran Associates Inc.

Mukherjee, A., Liu, B., & Glance, N. (2012). Spotting fake reviewer groups in consumer reviews. In *International conference on world wide web* (pp. 191–200). ACM.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., et al. (2013). *Spotting opinion spammers using behavioral footprints. KDD 2013* (pp. 632–640). New York: ACM Press.

Ott, M., Choi, Y., Cardie, C., & Hancock, J. T. (2011). Finding deceptive opinion spam by any stretch of the imagination. In *Meeting of the association for computational linguistics: Human language technologies: Vol. 1* (pp. 309–319).

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative deceptive opinion spam. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 497–501).

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 conference on empirical methods in natural language processing* (pp. 79–86).

Peng, Q., & Zhong, M. (2014). Detecting spam review through sentiment analysis. *Journal of Software, 9*(8), 2065–2072.

Rayana, S., & Akoglu, L. (2015). Collective opinion spam detection: Bridging review networks and metadata. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 985–994). ACM.

Ren, Y., Ji, D., & Zhang, H. (2014). Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 488–498).

Ren, Y., & Zhang, Y. (2016). Deceptive opinion spam detection using neural network. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 140–150).

Salton, G. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*(11), 613–620.

Savage, D., Zhang, X., Chou, P., et al. (2015). Detection of opinion spam based on anomalous rating deviation. *Expert Systems with Applications, 42*(22), 8650–8657.

Wang, G., Xie, S., Liu, B., & Yu, P. S. (2011). Review graph based online store review spammer detection. In *ICDM: 2011* (pp. 1242–1247).

Wang, X., Liu, K., He, S., et al. (2016). Learning to represent review with tensor decomposition for spam detection. In *EMNLP: 2016* (pp. 866–875).

Wu, G., Greene, D., & Smyth, B. (2010). Distortion as a validation criterion in the identification of suspicious reviews. In *The workshop on social media analytics* (pp. 10–13).

Xie, S., Wang, G., Lin, S., et al. (2012). Review spam detection via temporal pattern discovery. In *ACM SIGKDD international conference on knowledge discovery and data mining. KDD 2012* (pp. 823–831).

Xu, C., Zhang, J., Chang, K., et al. (2013). Uncovering collusive spammers in Chinese review websites. In *ACM international conference on conference on information & knowledge management. CIKM 2013* (pp. 979–988).

Xu, K., Ba, J., Kiros, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Computer science, ICML 2015* (pp. 2048–2057).

Xu, Y., Shi, B., Tian, W., et al. (2015). A unified model for unsupervised opinion spamming detection incorporating text generality. In *IJCAI 2015* (pp. 725–731).

Zhang, Y., Tan, Y., Zhang, M., Liu, Y., Tat-Seng, C., & Ma, S. (2015). Catch the black sheep: Unified framework for shilling attack detection based on fraudulent action propagation. In *International conference on artificial intelligence: 29* (pp. 2408–2414). AAAI Press.