Learning to Represent Review with Tensor Decomposition for Spam Detection

Xuepeng Wang[1,2], **Kang Liu**[1], **Shizhu He**[1] and **Jun Zhao**[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{xpwang, kliu, shizhu.he, jzhao}@nlpr.ia.ac.cn

Abstract

Review spam detection is a key task in opinion mining. To accomplish this type of detection, previous work has focused mainly on effectively representing fake and non-fake reviews with discriminative features, which are discovered or elaborately designed by experts or developers. This paper proposes a novel review spam detection method that learns the representation of reviews automatically instead of heavily relying on experts' knowledge in a data-driven manner. More specifically, according to 11 relations (generated automatically from two basic patterns) between reviewers and products, we employ tensor decomposition to learn the embeddings of the reviewers and products in a vector space. We collect relations between any two entities (reviewers and products), which results in much useful and global information. We concatenate the review text, the embeddings of the reviewer and the reviewed product as the representation of a review. Based on such representations, the classifier could identify the opinion spam more precisely. Experimental results on an open Yelp dataset show that our method could effectively enhance the spam detection accuracy compared with the state-of-the-art methods.

1 Introduction

With the development of E-commerce, more and more customers share their experiences about products and services by posting reviews on the web. These reviews could heavily guide the purchasing behaviors of customers. The products which receive more positive reviews tend to attract more consumers and result in more profits. Studies on Yelp.com have shown that an extra half-star rating could cause a restaurant to sell out 19% more products (Anderson and Magruder, 2012), and a one-star increase leads to a 5-9% profit increase (Luca, 2011). Therefore, more and more sellers and manufacturers have begun to place emphasis on analyzing reviews. However, the question remains: is every online review trustful? It has been reported that up to 25% of the reviews on Yelp.com could be fraudulent[1]. Due to the great profit or reputation, impostors or spammers energetically post fake reviews on the web to promote or defame targeted products (Jindal and Liu, 2008). Such fake reviews could mislead consumers and damage the online review websites' reputations. Therefore, it is necessary and urgent to detect fake reviews (review spam).

To accomplish this goal, much work has been conducted. They commonly regard this task as a classification task and most efforts are devoted to exploring useful features for representing target reviews. Li et al. (2013) and Kim et al. (2015) represent reviews with linguistic features; Lim et al. (2010) and Mukherjee et al. (2013c) represent reviews with reviewers' behavioral features[2]; Wang et al. (2011) and Akoglu et al. (2013) explore graph structure features[3]; Mukherjee et al. (2013b),

[1] http://www.bbc.com/news/technology-24299742

[2] Reviewers' spammer-like behaviors, e.g., if a reviewer continuously posts reviews within a short period of time, (s)he might be a spammer, and her (his) posted reviews could be spam.

[3] A kind of behavioral features which contain much interactions between reviewers and products

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 866–875,
Austin, Texas, November 1-5, 2016.

Rayana and Akoglu (2015) use the combination of aforementioned features. According to the existing studies, reviewers' behavioral features have been proven to be more effective than reviews' linguistic features for detecting review spam (Mukherjee et al., 2013c). It is because that foxy spammers could easily disguise their writing styles and forge reviews, discovering discriminative linguistic features is very difficult. Recently, most of the researchers (Rayana and Akoglu, 2015) have focused on the reviewers' behavioral features, the intuition behind which is to capture the reviewers' actions and supposes that those reviews written with spammer-like behaviors would be spam.

Although, the existing work has made significant progress in combating review spamming, they also have several limitations as follows. (1) The representations of reviews rely heavily on experts' prior knowledge or developers' ingenuity. To discover more discriminative features for representing reviews, previous work (Mukherjee et al., 2013b; Rayana and Akoglu, 2015) have spent lots of manpower and time on the statistics of the review datasets. Besides, experts' prior knowledge or developers' ingenuity is not always reliable with the variations of domains and languages. For example, based on the datasets from Dianping site[4], Li et al. (2015) find that the real users tend to review the restaurants nearby, but the spammers are not restricted to the geographical location, they may come from anywhere. However, it is not true in the Yelp datasets (Mukherjee et al., 2013b). We found that 72% of the Yelp's review spam is posted from the areas near the restaurants, but only 64% of the authentic reviews are near the restaurants. Therefore, how to learn the representations of reviews directly from data instead of heavily relying on the experts' prior knowledge or developers' ingenuity becomes crucial and urgent. (2) Furthermore, limited by the experts' knowledge, previous work only uses partial information of the review system. For example, traditional behavioral features (Lim et al., 2010; Mukherjee et al., 2013c) only utilize the information of individual reviewer. Although the work (Wang et al., 2011; Rayana and Akoglu, 2015) have tried to employ graph structure to consider the interactions among the reviewers and products, it is a kind of local interaction defined within the same product review page. However, the interaction among the reviewers and products from different review pages also provides much useful and global information, which is ignored by the previous work.

To tackle the problems described above, we propose a novel review spam detection method which can learn the representations of reviews instead of heavily relying on the experts' knowledge, developers' ingenuity, or spammer-like assumption, and can reserve the original information with a global manner. Inspired by the work about distributional representation or embedding for text and knowledge base, we propose a tensor factorization-based model to learn the representation of each review automatically. The finally learnt representation of each review is determined by the original data, rather than the features or clues found by experts. More specifically, we defined two basic patterns without any experts' knowledge, developers' ingenuity, or spammer-like assumptions. Based on the two basic patterns, we extended 11 interactive relations between entities (reviewers and products) in terms of time, locations, social contact, etc. Then, we build a 3-mode tensor on these 11 interactive relations between reviewers and products. In order to reserve the original information with a global manner, we collect the relations of any two entities regardless of whether they are from the same review page. In this way, we could reserve the original information of the data as much as possible, which dispenses with human selection. Next, we utilize tensor factorization to perform tensor decomposition, and the representations of reviewers and products are embedded in a latent vector space by collective learning. Afterward, we could obtain vector representations (embeddings) for both the reviewers and products. Then, we concatenate the review text (e.g., bigram), the embedding of a reviewer and the reviewed product as the representation of a review. In this way, the representations of reviews driven by data could be learnt in the entire review system in a global manner. Finally, such representations are fed into a classifier to detect the review spam.

In summary, this paper makes the following contributions:

- It addresses the spam detection issue with a

[4]http://www.dianping.com

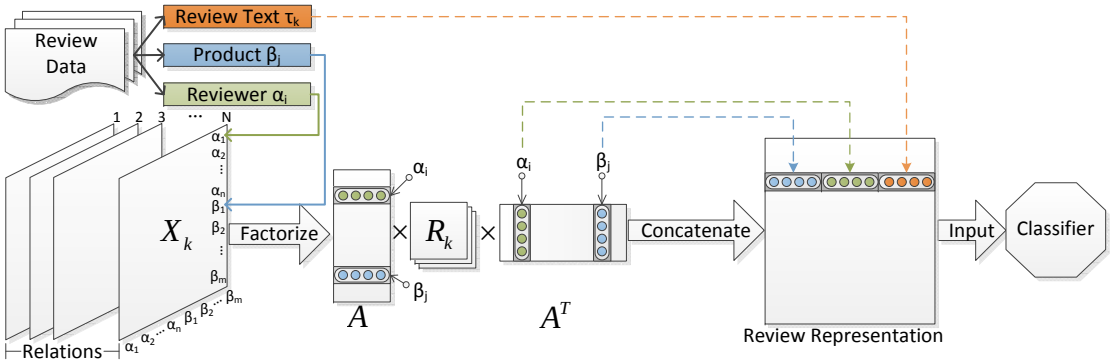


Figure 1: Illustrated of our method. The α_i denotes the i-th reviewer, and the β_j denotes the j-th product.

new perspective. Specifically, it learns the representation of reviews directly from the data. The key advantage is that it can represent the reviews instead of heavily relying on human ingenuity cost, experts' knowledge or any spammer-like assumption.

- It collects the relations between any two entities regardless of whether they are from the same review page, which results in much global information. With the help of tensor factorization, it could collectively embed the information of different relations into the final representations of reviews, and further optimize the representations. Therefore it could faithfully reflect the original characteristics of the entire review system with a global manner.
- An extra advantage is that the learnt representations of reviews are embeddings in a latent space. They are hardly comprehended by human beings included spammers. It's a robust detection method in contrast to the previous methods in which the reviews are represented by the explicit detecting clues and features. Once have realized the explicit features that were captured, experienced spammers could change their spamming strategies.
- The method of this paper renders 89.2% F1-score in detecting restaurant review spam which is higher than the F1-score of 86.1% rendered by the method in (Mukherjee et al., 2013b) (in hotel domain, it's 87.0% vs 84.8%). These experimental results give good confidence to the proposed approach, and the learnt representations of reviews are more robust and effective than in previous methods.

2 The Proposed Method

In this section, we propose our method (shown in Figure 1) in detail. Compared with the previous work, we address the review spam detection issue by learning the representation of the reviews automatically in a latent space without experts' knowledge. First, we extend 11 interactive relations between entities (reviewers and products) from the two basic patterns in terms of time, locations, social contact, etc. Then, our method generates 11 relation matrices of the reviewers (α_i) and products (β_j). After that, we construct a 3-mode tensor $\mathbf{X}$, where each slice X_k in $\mathbf{X}$ denotes the link relationship between the reviewers and products in the relation k. Second, we factorize the tensor $\mathbf{X}$ by employing the algorithm RESCAL (Nickel et al., 2011). In the factorization results, A represents the embeddings of the reviewers (α_i) and products (β_j) in the latent space with the collective learning. Third, we concatenate the review text (bigram), the embedding of its reviewer and the reviewed product together, as the representation of the review. Last, the concatenated embedding of the review is fed into a classifier (e.g., SVM) to detect whether it is a fake or non-fake review.

2.1 Relation Matrices Generation

In the review system, there are two kinds of entities: reviewers and products[5]. Each entity has several attributes, e.g., the attribute 'location' of a restaurant is Chicago (the restaurant is regarded as a product). More details are shown in Table 1.

To learn the representations of reviews directly from the data instead of experts' knowledge, we defined two basic patterns:

[5]The product refers to a hotel/restaurant in our experiments.

Reviewer Attribute	Product Attribute
set of reviewed products	set of reviewers
set of reviews (rating score, time)	set of reviews (rating score, time)
website joining date	average rating
friend count	review count
location	location

Table 1: Entities and Attributes

Pattern 1:Record the relationships between two entities.

Pattern 2:Record the relationships between attributes of two entities.

These patterns do not contain any spammer-like prior assumption, just record the natural relation in the original review system. Based on the two basic patterns, we extended 11 interactive relations between entities and their attributes (showed in Table 1). They will be described in detail as follows. Meanwhile, we define that $avg(a_{k,i}) = \frac{1}{n}\sum_{k=1}^{n} a_{k,i}$.

1. **Have reviewed:** This relation records whether a reviewer has reviewed a product. If reviewer α_i reviewed product β_j, the value $X[i,j,1]$ in this relation matrix $X[:,:,1]$ is 1, otherwise it's 0.

2. **Rating score:** What score (1 to 5 star) a reviewer-rated product receives. The value $X[i,j,2] \in \{1,2,...,5\}$.

3. **Commonly reviewed products:** The number of products that a reviewer commonly reviewed with other reviewers. The value $X[i,j,3] = |P_{ij}|, P_{ij} = P_i \cap P_j$; P_i is the product set reviewed by reviewer α_i.

4. **Commonly reviewed time difference:** The time differences that a reviewer who commonly reviews with other reviewers on the same products. The value $X[i,j,4] = \overline{t_i} - \overline{t_j}$, where $\overline{t_i} = avg(t_{k,i})$; $t_{k,i}$ is the time that the reviewer α_i reviewed the product β_k in the P_{ij} set.

5. **Commonly reviewed rating difference:** The rating differences that a reviewer who commonly reviews with other reviewers on the same products. The value $X[i,j,5] = \overline{r_i} - \overline{r_j}$, where $\overline{r_i} = avg(r_{k,i})$; $r_{k,i}$ is the score of the reviewer α_i rated the product β_k in P_{ij} set.

6. **Date difference of websites joined:** The date differences of joining review websites between a reviewer and others. The value $X[i,j,6] = d_i - d_j$, where d_i is the date on which reviewer α_i joining websites.

7. **Average rating difference:** The differences in the average rating of a reviewer over all his reviews compared with other reviewers. The value $X[i,j,7] = \overline{\gamma_i^r} - \overline{\gamma_j^r}$, where $\overline{\gamma_i^r} = avg(\gamma_{k,i}^r)$; $\gamma_{k,i}^r$ is the score with which the reviewer α_i rated the product β_k in P_i.

   The differences in the average rating of a product over all its reviews compared with other products. $X[i,j,7] = \overline{\gamma_i^p} - \overline{\gamma_j^p}$, where $\overline{\gamma_i^p} = avg(\gamma_{k,i}^p)$; $\gamma_{k,i}^p$ is the score of review k in R_i^β, which is the review set for product β_i.

8. **Friend count difference:** The differences in the friend count of a reviewer compared to others. At the review website, a reviewer can make friends with others. The value $X[i,j,8] = f_i - f_j$; where f_i is the friend count of reviewer α_i.

9. **Have the same location or not:** Whether two reviewers/products are from the same city or whether a reviewer has the same location with a product. If two entities have the same location, the value $X[i,j,9] = 1$, otherwise $X[i,j,9] = 0$.

10. **Common reviewers:** The number of the same reviewers that a product has with other products. The value $X[i,j,10] = |\Theta_{ij}|$, where $\Theta_{ij} = \Theta_i \cap \Theta_j$; Θ_i is the set of reviewers who reviewed product β_i.

11. **Review count difference:** The differences in the reviews count of any two reviewers. The value $X[i,j,11] = |R_i^\alpha| - \left|R_j^\alpha\right|$, where R_i^α is the reviews set of reviewer α_i. Or the differences in the reviews count of any two products, where $X[i,j,11] = \left|R_i^\beta\right| - \left|R_j^\beta\right|$, where R_i^β is the reviews set of product β_i.

According to the relations that we present above, we build 11 relation matrices among the reviewers and products. To unify the values of different matrices to a reference system, we normalize with the

sigmoid function. Thus, the value '0' will be normalized to '0.5'. Moreover, we set the values that make no sense to '0', such as the value between two products in Relation 1: Have reviewed. Then, we unite the 11 matrices to form the adjacent tensor. Each of the matrices is a slice of the tensor. The reviewers and products are regarded as the same entities in the tensor. We build two separate tensors for the hotel domain and restaurant domain respectively. Next, we perform tensor factorization to learn the representations (embeddings) of reviewers and products. Note that the word "relation" is normally used for binary (0/1) relations, but some values of aforementioned relations could be between 0 and 1. However, our experiments show that this type of relation is actually practicable. Besides, there is not any spammer-like assumption in the relations. Namely, the values of relations don't indicate how suspicious the reviewers are. The values faithfully reflect the original characteristics of the entire review system. This can help to reduce the need of carefully designing expert features and the understanding of domains as much as possible.

2.2 Learning to Represent Reviews

In general case, a review contains the text, the reviewer and the reviewed product. We firstly learn to represent reviewers and products. As mentioned above, based on the relations, we could construct an adjacency tensor $\mathbf{X}$. Then, we convert the global relation information related reviewers and products into embeddings through tensor factorization, where an efficient factorization algorithm called RESCAL (Nickel et al., 2011) is employed. First, we introduce it briefly.

To identify latent components in a tensor for collective learning, Nickel et al. (2011) proposed RESCAL, which is a tensor factorization algorithm. Given a tensor $X_{n' \times n' \times m'}$, RESCAL aims to have a rank-r approximation, where each slice X_k is factorized as

$$X_k \approx \boldsymbol{A} R_k \boldsymbol{A}^T, \text{ for all } k = 1...m', \qquad (1)$$

$\mathbf{A}$ is an $n' \times r$ matrix, where the i-th row denotes the i-th entity. R_k is an asymmetric $r \times r$ matrix that describes the interactions of the latent components according to the k-th relation. Note that while R_k differs in each slice, $\mathbf{A}$ remains the same.

$\mathbf{A}$ and R_k are derived by minimizing the loss function below.

$$\min_{\boldsymbol{A}, R_k} f(\boldsymbol{A}, R_k) + \lambda \cdot g(\boldsymbol{A}, R_k), \qquad (2)$$

where $f(\boldsymbol{A}, R_k) = \frac{1}{2}(\sum_k \parallel X_k - \boldsymbol{A} R_k \boldsymbol{A}^T \parallel_F^2)$ is the mean-squared reconstruction error, and $g(\boldsymbol{A}, R_k) = \frac{1}{2}(\parallel \boldsymbol{A} \parallel_F^2 + \sum_k \parallel R_k \parallel_F^2)$ is the regularization term.

In our method, slice X_k is the k-th relation above. The i-th entity is the i-th reviewer or product.

As mentioned in Section 2.1, in order to obtain more useful and global information automatically, we collect the relations of any two entities no matter whether they are from the same review page. Then we could embed the informations over multi-relations into the finally learnt representation by the tensor factorization. As Nickel et al. (2011) proved, all the relations have a determining influence on the learnt latent-component representation of the i-th entity. It removes the noise of the original data by learning through the global loss function. Consequently, we get the representation of reviewers and products with a further optimization by the collective learning.

2.3 Detecting Review Spam in Latent Space

After learning the representations of reviewers and products, we begin to represent the reviews that were written by reviewers for the products. Our final purpose is to detect the review spam. We concatenate the review text (bigram), the embedding of a reviewer and the reviewed product as the representation of a review. The representations of the review text by bigram have been proved to be effective in several previous work (Mukherjee et al., 2013b; Rayana and Akoglu, 2015; Kim et al., 2015). It's also a kind of data-driven representation. Then, we take the embeddings of the reviews as the input to the classifiers. Here, we use the linear kernel SVM model to compare with the experimental results in (Mukherjee et al., 2013b) and (Rayana and Akoglu, 2015).

3 Experiments

3.1 Datasets and Evaluation Metrics

Datasets: To evaluate the proposed method, we conducted experiments on Yelp dataset that was used in

previous studies (Mukherjee et al., 2013b; Mukherjee et al., 2013c; Rayana and Akoglu, 2015). Although there are other datasets for evaluation, such as (Jindal and Liu, 2008), (Lim et al., 2010; Xie et al., 2012) and (Ott et al., 2011), they are generated by human labeling or crowd sourcing and have been proved not to be reliable since human labeling fake reviews is quite poor (Ott et al., 2011). There was lack of real-life and nearly ground truth data, until Mukherjee et al. (2013c) proposed the Yelp review dataset. The statistics of the Yelp dataset are listed in Table 2. The reviewed product here refers to a hotel or restaurant.

Evaluation Metrics: We select precision (P), recall (R), F1-Score (F1) and accuracy (A) as metrics.

Domain	Hotel	Restaurant
fake	802	8368
non-fake	4876	50149
%fake	14.1%	14.3%
#reviews	5678	58517
#reviewers	5124	35593

Table 2: Yelp Labeled Dataset Statistics.

3.2 Our Method vs. The State-of-the-art Methods

To illustrate the effectiveness of the proposed approach, we select several state-of-the-arts for comparison. The first one is SPEAGLE$^+$ (Rayana and Akoglu, 2015), which is a kind of graph-based method. The representations of reviews in (Rayana and Akoglu, 2015) are combined with linguistic features, behavioral features and review graph structure features. It's a semi-supervised method. For a fair comparison with our 5-fold CV classification, we set the ratio of labeled data in SPEAGLE$^+$ to 80%. The second one is Mukherjee et al. (2013b). KC and Mukherjee (2016) also conduct experiments on the restaurant subset in Table 2. But they mainly focus on analyzing the effects of temporal dynamics. It's not our focus. So we didn't take it into comparison. In our experiments, we employ behavioral features (Mukherjee_BF) and both of behavioral and linguistic features (Mukherjee_BF+Bigram) proposed in Mukherjee et al. (2013b), respectively. The parameters used in these compared methods are same as the original papers. For our approach, we set the parameter r to 150, λ to 10, and the iteration number to 100.

The compared results are shown in Table 3. We utilize our learnt embeddings of reviewers (Ours_RE), both of reviewers' embeddings and products' embeddings (Ours_RE+PE), respectively. Moreover, to perform fair comparison, like Mukherjee et al. (2013b), we add representations of the review text in classifier (Ours_RE+PE+Bigram). From the results, we can observe that our method could outperform all state-of-the-arts in both the hotel and restaurant domains. It proves that our method is effective. Furthermore, the improvements in both the hotel and restaurant domains prove that our model possesses preferable domain-adaptability. It could represent the reviews more accurately and globally by learning from the original data, rather than the experts' knowledge or assumption.

3.3 The Effectiveness of Learning to Represent Review

To further prove the representations learnt by our method are effective for detecting review spam, we compare the learnt representation (embeddings) of reviewers (Ours_RE) (Table 3 (a,b) rows 7, 8) with existing behavioral features of reviewers (Mukherjee_BF) (Mukherjee et al., 2013b) (Table 3 (a,b) rows 3, 4). In results, using the learnt reviewers' representations in our method, results in around 2.0% (in 50:50) and 4.0% (in N.D.) improvement in F1 and A in the hotel domain, and results in around 2.1% (in 50:50) and 7.0%(in N.D.) improvement in F1 and A in the restaurant domain. These results show that our data-driven representations of reviewers are more helpful for review spam detection than existing reviewers' behavioral features, and that new method embeds more useful and accurate information from the original data. It isn't limited to experts' knowledge. Moreover, the latent representations are more robust because they are hardly perceived by spammers. Having realized the explicit existing behavioral features, crafty spammers tend to change their spamming strategies. Consider the feature "Review Length", which is used in (Mukherjee et al., 2013b), as an example. They find that the average review length of the spammers is quite short compared with non-spammers. However, once a crafty spammer realizes that he left this type of footprint, he could produce a review that is as long as the non-

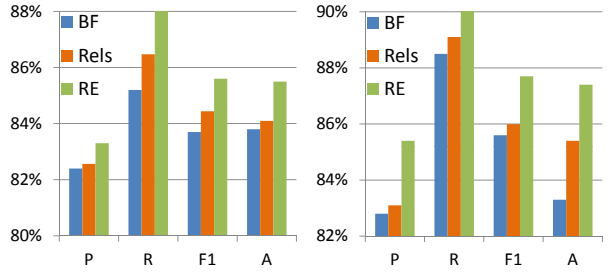
Method	C.D.	P	R	F1	A
SPEAGLE$^+$(80%)	50:50	75.7	83.0	79.1	81.0
	N.D.	26.5	56.0	36.0	80.4
Mukherjee_BF	50:50	82.4	85.2	83.7	83.8
	N.D.	41.4	84.6	55.6	82.4
Mukherjee_BF+Bigram	50:50	82.8	86.9	84.8	85.1
	N.D.	46.5	82.5	59.4	84.9
Ours_RE	50:50	83.3	88.1	**85.6**	**85.5**
	N.D.	47.1	83.5	**60.2**	**85.0**
Ours_RE+PE	50:50	83.6	89.0	**86.2**	**85.7**
	N.D.	47.5	84.1	**60.7**	**85.3**
Ours_RE+PE+Bigram	50:50	84.2	89.9	**87.0**	**86.5**
	N.D.	48.2	85.0	**61.5**	**85.9**

(a) Hotel

P	R	F1	A	
80.5	83.2	81.8	82.5	1
50.1	70.5	58.6	82.0	2
82.8	88.5	85.6	83.3	3
48.2	87.9	62.3	78.6	4
84.5	87.8	86.1	86.5	5
48.9	87.3	62.7	82.3	6
85.4	90.2	**87.7**	**87.4**	7
56.9	90.1	**69.8**	**85.8**	8
86.0	90.7	**88.3**	**88.0**	9
57.4	89.9	**70.1**	**86.1**	10
86.8	91.8	**89.2**	**89.9**	11
58.2	90.3	**70.8**	**87.8**	12

(b) Restaurant

Table 3: Classification results across the behavioral features (BF), the reviewer embeddings (RE) , product embeddings (PE) and bigram of the review texts. Training uses balanced data (50:50). Testing uses two class distributions (C.D.): 50:50 (balanced) and Natural Distribution (N.D.). Improvements of our method are statistically significant with p<0.005 based on paired *t*-test.



(a) Hotel **(b)** Restaurant

Figure 2: SVM 5-fold CV classification results across behavioral features (BF), 11 relations (Rels) and reviewer embeddings (RE) in our method. Both training and testing use balanced data (50:50). Improvements are statistically significant with p<0.005 based on paired *t*-test.

Dropped Relation	Hotel		Restaurant	
	F1	A	F1	A
1	-2.1	-2.0	-2.0	-3.1
2	-2.3	-2.1	-1.9	-2.9
3	-3.9	-4.0	-4.0	-6.3
4	-3.7	-3.5	-3.6	-5.5
5	-3.5	-3.6	-2.8	-4.5
6	-2.5	-2.5	-3.4	-5.2
7	-3.2	-3.2	-3.3	-5.0
8	-2.8	-2.6	-3.0	-4.6
9	-4.0	-3.7	-3.7	-5.4
10	-2.2	-2.4	-1.8	-2.8
11	-2.6	-2.4	-2.7	-4.4

Table 4: SVM 5-fold CV classification results by dropping relations from our method utilizing RE+PE+Bigram. Both training and testing use balanced data (50:50). Differences in classification metrics for each dropped relation are statistically significant with p<0.01 based on paired *t*-test.

spammers to pretend to be a normal reviewer. Besides, as there isn't any spammer-like assumption in our extended relations (Section 2.1), crafty spammers have little influence on them.

We also compared existing behavioral features (BF) (Mukherjee et al., 2013b) with detecting review spam by only employing the 11 generated relations (Rels). We take the relation matrix row of each reviewer as the representations of the reviews. According to the results shown in Figure 2, the 11 generated relations (Rels) results in an obvious improvement than the existing behavioral features (BF) (Mukherjee et al., 2013b) (Table 3 (a,b) row 3) in both the hotel and restaurant domains. It proves that the generated relations could obtain more useful and global informations, as they collect the relations of any entities (reviewers and products) regardless of whether they are from the same review page. Furthermore, Figure 2 also showed that the embeddings of reviewers (RE) learnt by the tensor decomposition perform better than the Rels. As we mentioned in Section 2.2, the tensor decomposition embeds the informations over all the relations collectively, and removes the noise of the original data by learning through the global loss function. Consequently, we get the representations with a further optimization.

3.4 The Effectiveness of Product Embeddings

In general case, a review contains the review text, the reviewer and the reviewed product. But most of the previous work represent the reviews with the reviewers' behavioral features and the reviews' linguistic features. The products are seldom represented. As shown in Table 3 (a,b) rows 9,10 , the representations which added the products embeddings

perform better than just using the reviewer embeddings. Statistics of the datasets suggest that there are about 1% of spammers who not only write fake reviews, but also write non-fake reviews. Liu (2015) also proved that some reviewers have contributed many genuine reviews and have built up their reputation; then they started to spam for some businesses, or even sell their accounts to spammers. Compared with previous work, our method by adding product embeddings could distinguish the reviews of the same reviewer for different products.

3.5 The Effects of Different Relations

We also drop relations of our method with a graceful degradation. Table 4 shows the performances of our method utilizing BF+PE+Bigram for hotel and restaurant domains. We found that dropping Relations 1, 2 and 10 results in a relatively gentle reduction (about 2.2%) in F1-score. According to our survey, the sparseness of the slices generated by Relation 1, 2 and 10 is about 99.9%. For this reason, the result is a relatively gentle reduction. Dropping other relations also result in a 2.5-4.0% performance reduction. It proves that each relation has an influence on the learning to represent reviews.

4 Related Work

Jindal and Liu (2008) first propose the problem of review spam detection. They identify three categories of spam: fake reviews (also called untruthful opinions), reviews on the brand only, and non-reviews.Stepping studies focus on studying fake reviews because of its difficulty to be detected. Most efforts are devoted to represent fake and non-fake reviews with effective features.

Linguistic Features Ott et al. (2011) apply psychological and linguistic clues to identify review spam. They produce the first dataset of gold-standard deceptive review spam, employing crowdsourcing through the Amazon Mechanical Turk. Harris (2012) explores several human- and machine-based assessment methods with writing style features. Feng et al. (2012a) investigate syntactic stylometry for review spam detection. Li et al. (2013) propose a generative LDA-based topic modeling approach for fake review detection. They (Li et al., 2014b) further investigate the general difference of language usage between deceptive and truthful reviews. Li et al. (2014a) propose a positive-unlabeled learning method base on unigrams and bigrams. Kim et al. (2015) carry out a frame-based deep semantic analysis on deceptive opinions.

Behavioral Features Lim et al. (2010) investigate reviewers' rating behavioral features. Jindal et al. (2010) identify unusual review patterns which can represent suspicious behaviors of reviews. Li et al. (2011) provide a two-view semi-supervised method, co-training method base on behavioral features. Feng et al. (2012b) study the distributions of behavioral features. Xie et al. (2012) explore the singleton reviews with abnormal temporal patterns. Mukherjee et al. (2012) study the group spammers' behavioral features. Mukherjee et al. (2013a) propose a principal method which models the spamicity of reviewers. Fei et al. (2013) model the reviewers' co-occurrence in review bursts. Mukherjee et al. (2013c) prove that reviewers' behavioral features are more effective than reviews' linguistic features for detecting review spam. Li et al. (2015) explore the temporal and spatial patterns at Dianping.com. KC and Mukherjee (2016) analyze the temporal dynamics of opinion spamming.

Graph Structure Wang et al. (2011) investigate the review graph features of online store review. Akoglu et al. (2013) exploit the network effect among reviewers and products.

Combined Features There are also some work which explores methods via the combined features referred above. Mukherjee et al. (2013b) prose a method base on the linguistic features and behavioral features. Rayana and Akoglu (2015) propose a model that utilizes clues from review text, reviewers' behaviors and the review graph structure.

5 Conclusion and Future Work

This paper proposes a new review spam detection method that learns the representations of reviews instead of heavily relying on experts' knowledge in a data-driven manner. A 3-mode tensor is built on the relations which are generated from two patterns, and a tensor factorization algorithm is used to automatically learn the vector representations of reviewers and products. Afterwards, we concatenate the review text, the embedding of a reviewer and the

reviewed product as the representation of a review. Then, a classifier is applied to detect the review spam. Experimental results prove the effectiveness of the proposed method, which learns more robust review representations. In future work, we plan to explore a more effective way to learn the embeddings of review text.

Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 61533018), the National Basic Research Program of China (No. 2014CB340503) and the National Natural Science Foundation of China (No. 61502493). We would like to thank Prof. Bing Liu for sharing the Yelp review dataset with us, and the anonymous reviewers for their detailed comments and suggestions.

References

Leman Akoglu, Rishi Chandy, and Christos Faloutsos. 2013. Opinion fraud detection in online reviews by network effects. *ICWSM*, 13:2–11.

Michael Anderson and Jeremy Magruder. 2012. Learning from the crowd: Regression discontinuity estimates of the effects of an online review database*. *The Economic Journal*, 122(563):957–989.

Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Exploiting burstiness in reviews for review spammer detection. In *ICWSM*. Citeseer.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012a. Syntactic stylometry for deception detection. In *Proceedings of the 50th ACL: Short Papers-Volume 2*, pages 171–175. ACL.

Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. 2012b. Distributional footprints of deceptive product reviews. In *ICWSM*.

C Harris. 2012. Detecting deceptive opinion spam using human computation. In *Workshops at AAAI on Artificial Intelligence*.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the First WSDM*, pages 219–230. ACM.

Nitin Jindal, Bing Liu, and Ee-Peng Lim. 2010. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th CIKM*, pages 1549–1552. ACM.

Santosh KC and Arjun Mukherjee. 2016. On the temporal dynamics of opinion spamming: Case studies on yelp. In *Proceedings of the 25th International Conference on World Wide Web*, pages 369–379. International World Wide Web Conferences Steering Committee.

Seongsoon Kim, Hyeokyoon Chang, Seongwoon Lee, Minhwan Yu, and Jaewoo Kang. 2015. Deep semantic frame-based deceptive opinion spam analysis. In *Proceedings of the 24th CIKM*, pages 1131–1140. ACM.

Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Learning to identify review spam. In *IJCAI Proceedings*, volume 22, page 2488.

Jiwei Li, Claire Cardie, and Sujian Li. 2013. Topicspam: a topic-model based approach for spam detection. In *ACL (2)*, pages 217–221.

Huayi Li, Bing Liu, Arjun Mukherjee, and Jidong Shao. 2014a. Spotting fake reviews using positive-unlabeled learning. *Computación y Sistemas*, 18(3):467–475.

Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. 2014b. Towards a general rule for identifying deceptive opinion spam. ACL.

Huayi Li, Zhiyuan Chen, Arjun Mukherjee, Bing Liu, and Jidong Shao. 2015. Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In *Ninth International AAAI Conference on Web and Social Media*.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th CIKM*, pages 939–948. ACM.

Bing Liu. 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press.

Michael Luca. 2011. Reviews, reputation, and revenue: The case of yelp. com. *Com (September 16, 2011). Harvard Business School NOM Unit Working Paper*, (12-016).

Arjun Mukherjee, Bing Liu, and Natalie Glance. 2012. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st WWW*, pages 191–200. ACM.

Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013a. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD*, pages 632–640. ACM.

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. 2013b. Fake review detection: Classification and analysis of real and pseudo reviews. Technical report, Technical Report UIC-CS-2013-03, University of Illinois at Chicago.

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S Glance. 2013c. What yelp fake review filter might be doing? In *ICWSM*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective

learning on multi-relational data. In *Proceedings of the 28th ICML*, pages 809–816.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th ACL: Human Language Technologies-Volume 1*, pages 309–319. ACL.

Shebuti Rayana and Leman Akoglu. 2015. Collective opinion spam detection: Bridging review networks and metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 985–994. ACM.

Guan Wang, Sihong Xie, Bing Liu, and Philip S Yu. 2011. Review graph based online store review spammer detection. In *Proceedings of the 11th ICDM*, pages 1242–1247. IEEE.

Sihong Xie, Guan Wang, Shuyang Lin, and Philip S Yu. 2012. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th KDD*, pages 823–831. ACM.