Information Sciences 385–386 (2017) 213–224

Contents lists available at ScienceDirect

Information Sciences

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

Neural networks for deceptive opinion spam detection: An empirical study

Yafeng Ren[a,*], Donghong Ji[a,b]

[a] Guangdong Collaborative Innovation Center for Language Research & Services, Guangdong University of Foreign Studies, Guangzhou 510420, China
[b] Computer School, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Article history:
Received 11 March 2016
Revised 29 December 2016
Accepted 2 January 2017
Available online 3 January 2017

Keywords:
Deceptive opinion spam
Discrete features
Convolutional neural network
Recurrent neural network
Representation learning

ABSTRACT

The products reviews are increasingly used by individuals and organizations for purchase and business decisions. Driven by the desire of profit, spammers produce synthesized reviews to promote some products or demote competitors products. So deceptive opinion spam detection has attracted significant attention from both business and research communities in recent years. Existing approaches mainly focus on traditional discrete features, which are based on linguistic and psychological cues. However, these methods fail to encode the semantic meaning of a document from the discourse perspective, which limits the performance. In this work, we empirically explore a neural network model to learn document-level representation for detecting deceptive opinion spam. First, the model learns sentence representation with convolutional neural network. Then, sentence representations are combined using a gated recurrent neural network, which can model discourse information and yield a document vector. Finally, the document representations are directly used as features to identify deceptive opinion spam. Based on three domains datasets, the results on in-domain and cross-domain experiments show that our proposed method outperforms state-of-the-art methods.

1. Introduction

In recent years, online reviews on products and services contain rich information related to subjective opinions on certain topics. These information have become an important resource for public opinion that influence our decisions over an extremely wide spectrum of daily and professional activities: e.g.,where to eat, where to stay, which products to purchase, which doctors to see, and so on. As a result, sentiment analysis and opinion mining based on product reviews have become a heated topic in natural language processing (NLP) [6,11,41].

Since reviews information can guide people purchase behavior, positive reviews can result in huge economic benefit and fame for organizations or individuals. This gives powerful incentive to promote the generation of deceptive opinion spam [24,28,39,47]. Deceptive opinion spam is a type of review with fictitious opinions, deliberately written to sound authentic [21,34]. Two reviews are shown as follows:

[*] Corresponding author.
E-mail address: renyafeng@whu.edu.cn (Y. Ren).

http://dx.doi.org/10.1016/j.ins.2017.01.015
0020-0255/© 2017 Elsevier Inc. All rights reserved.

- *I have stayed at many hotels travelling for both business and pleasure and I can honestly stay that the James is tops. The service at the hotel is first class. The rooms are modern and very comfortable. The location is perfect within walking distance to all the great sights and restaurants. Highly recommend to both business travellers and couples. (Date of review: Jun 9, 2006)*
- *My husband and I stayed at the James Chicago Hotel for our anniversary. This place is fantastic! We knew as soon as we arrived we made the right choice! The rooms are BEAUTIFUL and the staff very attentive and wonderful!! The area of the hotel is great, since I love to shop I couldn't ask for more!! We will definatly be back to Chicago and we will for sure be back to the James Chicago. (Date of review: Jun 9, 2006)*

These two reviews are from the firstly public dataset in the domain of opinion spam [34]. The first is non-spam or truthful review, and the second is deceptive opinion spam. Based on two reviews, we can know that it is very difficult for human readers to distinguish them from truthful reviews. In a test by previous work [34], the average accuracy of three human judges is only 57.33%. Deceptive opinion spam detection is a pressing and also profound issue as it is critical to ensure that trustworthiness of the information on the web. Without detecting them, the social media could become a place full of lies, fakes, and deceptions and completely useless. Hence, machine learning methods for automatically detecting deceptive opinion spam can be very necessary.

Generally, deceptive opinion spam detection is deemed to be a classification problem [34,39]. Based on the positive and negative examples annotated by people, supervised learning is utilized to build a classifier, and then an unlabeled review can be predicted as deceptive review or truthful one. So the objective of the task is to identify whether a given document a spam or not. The majority of existing approaches follow the seminal work of Jindal and Liu (2008) [21], employing classifiers with supervised learning. Most studies focus on designing effective features to enhance classification performance. Typical features represent linguistic and psychological cues, but fail to effectively represent a document from the viewpoint of global discourse structures. For example, Ott et al. (2011) and Li et al. (2014) represent documents with Unigram, POS and LIWC (Linguistic Inquiry and Word Count) feature [17,34]. Although such features give the strong performance, their sparsity makes it difficult to capture non-local semantic information over a sentence or discourse.

Recently, neural network models have been used to learn semantic representations for NLP tasks [16,50], achieving highly competitive results. The potential advantages of neural networks for spam detection are three-fold. First, neural models use real-valued hidden layers for automatic feature combinations, which can capture complex global semantic information that is difficult to express using traditional discrete manual features. This can be useful in addressing the limitation of discrete models mentioned above. Second, neural networks take distributed word embeddings as inputs, which can be trained from large-scale raw text, thus alleviating the scarcity of annotated data to some extent. Third, neural network models can learn continuous document representations, leveraging sentence and discourse models simultaneously.

In this paper, we show that significant improvements can be achieved by learning continuous document representations using a neural network model. In particular, we propose a three-stage system for opinion spam detection, as shown in Fig. 1. In the first stage, a convolutional neural network is used to produce sentence representations from word representations. Then a bi-directional gated recurrent neural network is used to construct a document representation from the sentence vectors by modeling their semantic and discourse relations. Finally, the document representation is used as features to identify deceptive opinion spam. Such automatically induced dense document representation is compared with traditional manually-designed features for the task.

We evaluate the proposed model on a standard benchmark [17], which consists of data from three domain (*Hotel, Restaurant*, and *Doctor*). Results on in-domain and cross-domain experiments show that our proposed neural model significantly outperforms the state-of-the-art methods, demonstrating the advantage of neural models in capturing semantic characteristics.

In remaining parts, Section 2 presents related work. Section 3 gives details of our proposed neural model. Section 4 introduces experimental setup, and then reports experimental results of in-domain and cross-domain settings. Section 5 concludes this work.

2. Related work

2.1. Deceptive opinion spam detection

Spam detection has been historically investigated in the Web-page and E-mail domains [8,33,55]. With the rise of e-commerse, spam detection research has recently been extended to the customer review domain [17,31,34]. Various types of indicator features have been studied. Jindal and Liu (2008) first studied deceptive opinion spam problem, training models using features based on the review content, reviewer, and the product itself [21]. Yoo and Gretzel (2009) gathered 40 truthful and 42 deceptive hotel reviews and manually compared the linguistic differences between them [54].

Ott et al. (2011) created a benchmark dataset by employing *Turkers* to write fake reviews [34]. Their data was adopted by a line of subsequent work [9,10,35]. For example, Feng et al. (2012) looked into syntactic features from context free grammar parse trees to improve the classification performance [9]. Feng and Hirst (2013) built profiles of hotels from collections of reviews, measuring the compatibility of customer reviews to the hotel profile, and using it as a feature for opinion spam detection [10]. Newman et al. (2003) looked into some linguistic cues to deception detection, such as increased negative emotion terms and decreased spatial detail [32]. They found certain writing style difference between informative and imag-

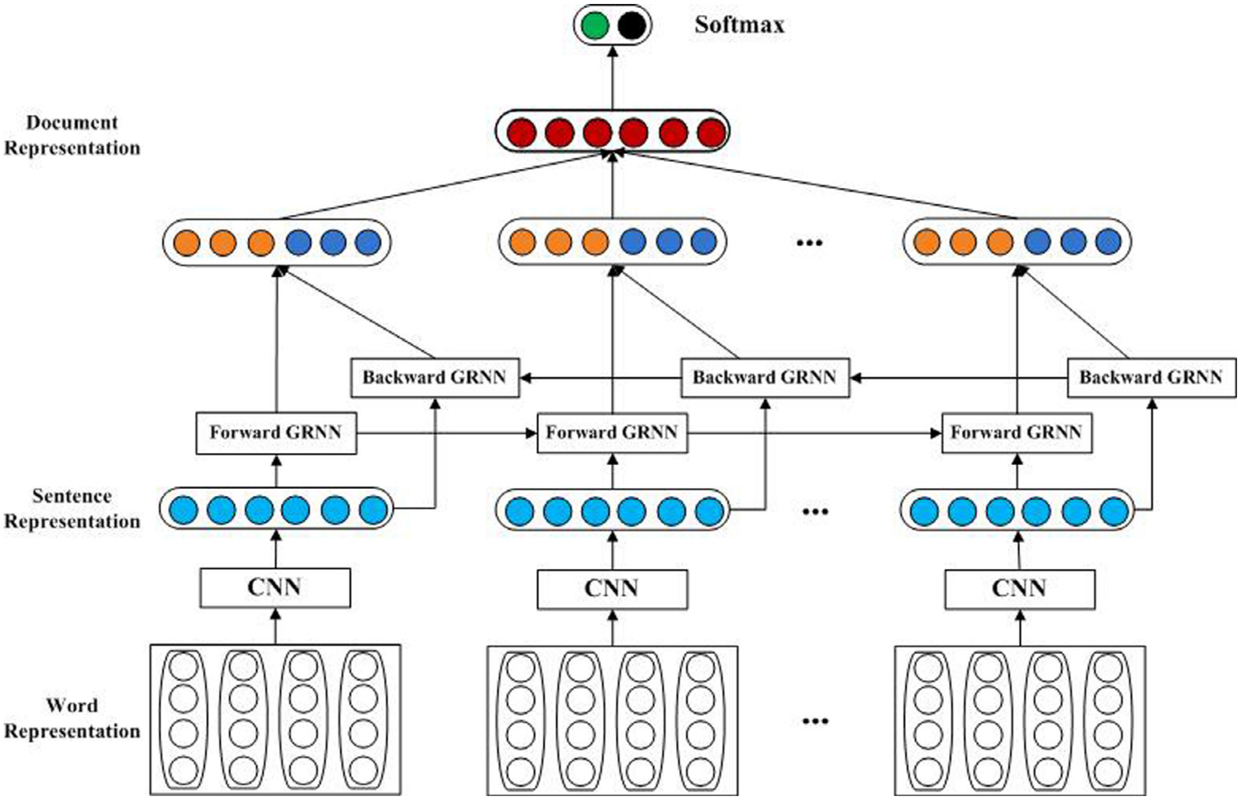

Fig. 1. Neural network model for deceptive opinion spam detection.

inative texts. Recently, Li et al. (2014) created a wider-coverage benchmark dataset [17], which comprises of data from three domains (*Hotel, Restaurant*, and *Doctor*), and explored generalized approaches for identifying online deceptive opinion spam. We adopt this dataset for our experiments due to its larger size and coverage.

Existing methods use traditional discrete features, which can be sparse and fail to effectively encode the semantic information from the overall discourse. In this paper, we propose to learn document-level neural representation for better detecting deceptive opinion spam.

There has been work that exploit features outside the review content itself. In additon to Jindal and Liu (2008), Mukherjee et al. (2013) explored the features from customer's behavior to identify deception [21,31]. Qian and Liu (2013) identified multiple user IDs that are generated by the same author, as these authors are more likely to generate deceptive reviews [38]. Ren et al. (2014) and Rout et al. (2016) proposed a semi-supervised learning method, and built an accurate classifier to identify deceptive reviews [39,44]. Besides, Ren et al. (2015) presented a novel approach, from the viewpoint of correcting the mislabeled instances, to find deceptive opinion spam [40]. Kim et al. (2015) introduced a frame-based semantic feature based on FrameNet, and experimental results showed that semantic frame features could improve the classification accuracy [26]. We focus on the review content in this paper, but their features can be used to extend our model.

2.2. Neural models for representation learning

Recently, neural networks have been exploited to learn continuous representation for a variety of NLP tasks [5,23,42]. Distributed word representations [29,37,43] have been used as the basic building block by most models for NLP. Numerous methods have been proposed to learn representations of phrases and larger text segments from distributed word representations. For example, Yessenalina and Cardie (2011) use iterated matrix multiplication to derive phrase representations from word representations [53]. Le and Mikolov (2014) introduce paragraph vector to learn document representations [16]. Socher et al. (2013) introduced a family of recursive neural networks to represent sentence-level semantic composition [46]. Later, this work is extended by different aspects, which contains global feed backward mechanisms [36], deep recursive layers [15], feature weight tuning [18], adaptive composition functions [7] and combinatory categorial grammer [13].

Convolutional neural networks (CNN) have been widely used for semantic composition [22,23], automatically capturing n-gram information. Sequential model such as recurrent neural network (RNN) or long short-term memory (LSTM) have also been used for recurrent semantic composition [19,50]. We empirically explore convolutional neural networks and recurrent neural networks to learn document representation for detecting deceptive opinion spam, comparing their effect with bag-of-word and paragraph vector baselines.

3. Methodology

The proposed neural network model learns continuous vector representations for documents of variable lengths, which is used as features to classify a spam or not for each document. Shown in Fig. 1, it mainly consists of two components. The first component produces continuous sentence vector representations from word representations (Section 3.1), and the second component takes sentence vectors as inputs, giving document representations (Section 3.2).

Structurally, the composition of words in forming sentences is similar to the composition of sentences in forming documents, both tracking sequences of inputs with long range dependencies. Both CNN and RNN are typically used for representing sequences in NLP, giving state-of-the-art accuracies in various tasks. For example, for modeling sentences, CNN gives the best results for sentiment analysis [22,43], while LSTM gives the best results for question answering [52]. For modeling discourse structures, LSTM has been used for more frequently [20]. We experimented with both CNN and RNN for both sentence and document modeling, finding that the best development accuracies are obtained when CNN is used for sentence modeling and RNN is used for document modeling. Therefore, we choose this structure in Fig. 1. Note, however, that our main goal is to empirically study the effectiveness of neural features in contrast to manual discrete features, rather than find a most accurate neural model variation for this task.

3.1. Sentence model

We represent the word using embeddings [2], which are low dimensional, continuous and real-valued vectors. For each word w, we use a look-up matrix $\mathbf{E}$ to obtain its embedding $e(w) \in R^D$, where $\mathbf{E} \in R^{D\times V}$ is a model parameter, D is the word vector dimension and V is the vocabulary size. In a typical neural model, $\mathbf{E}$ can be randomly initialized from a uniform distribution [46], or pre-trained from a large raw corpus with embedding learning algorithm [29]. We study the effect of initialization in our experiments.

To model semantic representations of sentences, convolutional neural network (CNN) and recursive neural network [46] are two state-of-the-art methods. The convolution action has been commonly used to synthesize lexical n-gram information [5,45]. N-grams have been shown useful for many NLP tasks [30,43,49], and we apply them to our neural network. As shown in the bottom of Fig. 1, a convolutional neural network [22,23,25] is used to learn continuous representations of a sentence as it does not rely on external parse tree. Specifically, we use three convolutional filters to produce sentence representation. The reason is that they are capable of capturing local semantics of n-grams of various granularities, including unigram, bigrams and trigrams, respectively. This is proven powerful for some NLP task, such as sentiment classification [42]. Fig. 2 illustrates the CNN with three convolutional filters. Formally, denote a sentence consisting of n words as $\{w_1, w_2, .., w_i, ..w_n\}$. Each word w_i is mapped to the embedding representation $e(w_i) \in R^D$. A convolutional filter is a list of linear layers with shared parameters. Let D_1, D_2, D_3 be the width of the three convolutional filters, respectively. Taking D_1 for example, W_1 and b_1 are the shared parameters of linear layers for this filter. The input of a linear layer is the concatenation of word embeddings in a fixed-length window size D_1, which is denoted as $I_{1,i} = [e(w_i); e(w_{i+1}); \ldots; e(w_{i+D_1-1})] \in R^{D\times D_1}$. The output of a linear layer is calculated as

$$H_{1,i} = W_1 \cdot I_{1,i} + b_1, \tag{1}$$

where $W_1 \in R^{l_{oc}\times D\times D_1}$, l_{oc} is the output size of linear layer. We use an average pooling layer to merge the varying number of outputs $\{H_{1,\,1}, H_{1,\,2}, .., H_{1,\,n}\}$ from convolution layer into a vector with fixed dimensions.

$$H_1 = \frac{1}{n}\sum_{i=1}^{n} H_{1,i} \tag{2}$$

To incorporate nonlinearity, a activation function *tanh* is used to obtain the output O_1 of this filter.

$$O_1 = tanh(H_1) \tag{3}$$

Similarly, we obtain the O_2 and O_3 for the other two convolutional filters with width 2 and 3, respectively. To capture the semantics of a sentence, we average the outputs of three filters to generate sentence representation.

$$s = \frac{1}{3}(O_1 + O_2 + O_3) \tag{4}$$

3.2. Document model

Given a document with m sentences, we use the sentence vectors $s_1, s_2, .., s_m$ obtained by CNN as inputs, and learn document composition with a gated recurrent neural network (GRNN). Various methods can be used to this end. For example, a simple method is to ignore the order of sentences and average the input sentence vectors as a document vector. However, this method fails to capture semantic relations (e.g. “contrast” and “cause”) between sentences. CNN is an other alternative

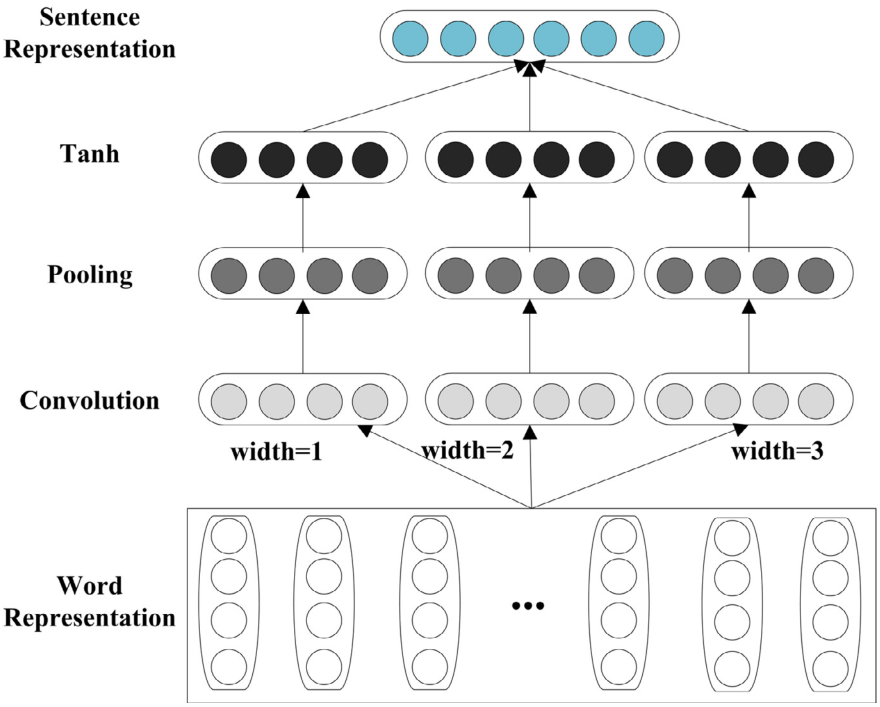

Fig. 2. Convolutional neural network with three different width for sentence representation.

for document composition, which models local sentence relations with shared parameters of linear layers. However, CNN does not directly model long-range discourse structures, which is crucial to represent a document.

Standard recurrent neural networks (RNN) map sentence vectors of variable lengths to a fixed-length vector, by starting with a initial vector, and recurrently transforming the current sentence vector s_t together with the previous state vector h_{t-1} into a new state vector h_t. The transition function is typically a linear layer followed by a non-linear activation function such as *tanh*

$$h_t = tanh(W_r \cdot [h_{t-1}; s_t] + b_r), \tag{5}$$

where $W_r \in R^{l_h \times (l_h + l_{oc})}$, $b_r \in R^{l_h}$, l_h and l_{oc} are dimensions of state vectors and sentence vectors, respectively. Unfortunately, the standard RNN suffers the problem of vanishing gradients [1,14]. This makes it difficult to model long-distance correlation in a sequence. To address this problem, we explore a gated recurrent neural network (GRNN) for document composition. The approach is analogous to LSTM [3,4], but empirically runs faster. Specifically, the transition function of the GRNN used in the work is calculated as follows.

$$i_t = sigmoid(W_i \cdot [h_{t-1}; s_t] + b_i) \tag{6}$$

$$f_t = sigmoid(W_f \cdot [h_{t-1}; s_t] + b_f) \tag{7}$$

$$g_t = tanh(W_r \cdot [h_{t-1}; s_t] + b_r) \tag{8}$$

$$h_t = tanh(i_t \odot g_t + f_t \odot h_{t-1}) \tag{9}$$

where $\odot$ stands for element-wise multiplication, i_t and f_t represent the reset gate and update gate, respectively. W_i, W_f, b_i, b_f adaptively select and remove history state vectors and input vectors for semantic composition. Fig. 3 summarizes the structure of our GRNN.

To capture discourse relations, we apply the GRNN structure over sentence representation vectors in the left-to-right and right-to-left directions, respectively, resulting in a forward state sequence h_1, h_2, .., h_n and a backward state sequence $h_n^{'}, h_{n-1}^{'}, .., h_1^{'}$, respectively. For each sentence vector node s_i, a combination of h_i and $h_i^{'}$ is used as its bi-directional state vector.

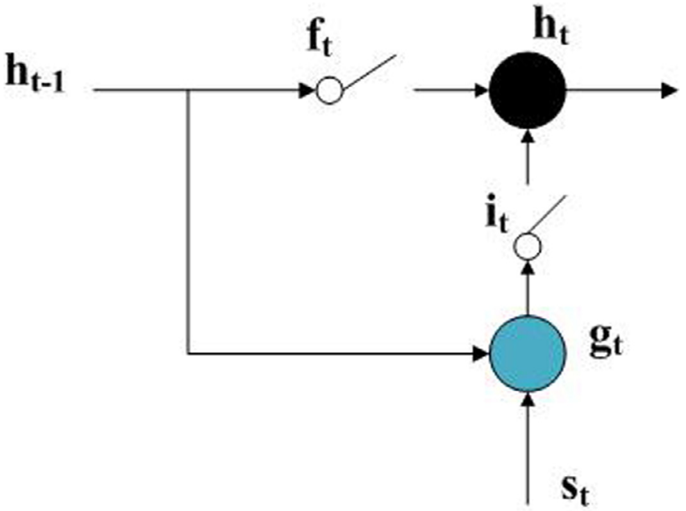

Fig. 3. The GRNN cell structure.

3.3. The classification model

We use the averaged state vector, which is the combination of h_i and $h_i^{'}$ from each sentence, as the document representation, which serves as features for identifying deceptive opinion spam. Specifically, a linear layer is added to transform the document vector into a real-valued vector v_i, whose length is class number C. A *softmax* function is added to convert real vector values to conditional probabilities, which is calculated as follows

$$P_i = \frac{exp(v_i)}{\sum_{i'=1}^{C} exp(v_{i'})} \tag{10}$$

In this way, the neural model avoids the need of manually defining features.

Our training objective is to minimize the cross-entropy loss over a set of training examples $(x_i, y_i)|_{i=1}^{N}$, plus a l_2-regularization term,

$$L(\theta) = -\sum_{i=1}^{N} \log \frac{e^{\vec{o}(y_i)}}{e^{\vec{o}(0)} + e^{\vec{o}(1)}} + \frac{\lambda}{2} \parallel \theta \parallel^2, \tag{11}$$

where θ is the set of model parameters, including $\mathbf{W}_1$, $\mathbf{b}_1$, $\mathbf{W}_2$, $\mathbf{b}_2$, $\mathbf{W}_3$, $\mathbf{b}_3$, $\mathbf{W}_i$, $\mathbf{b}_i$, $\mathbf{W}_f$, $\mathbf{b}_f$, $\mathbf{W}_r$, $\mathbf{b}_r$, $\mathbf{W}_{softmax}$ and $\mathbf{b}_{softmax}$.

We use online AdaGrad to minimize the objective. At step t, the parameters are updated by:

$$\theta_{t,i} = \theta_{t-1,i} - \frac{\alpha}{\sqrt{\sum_{t'=1}^{t} g_{t',i}^2}} g_{t,i}, \tag{12}$$

where α is the initial learning rate, and $g_{t,\ i}$ is the gradient of the ith dimension at step t.

We follow Glorot et al. (2011) and initialize all the matrix and vector parameters with uniform samples in $(-\sqrt{6/(r+c)}, -\sqrt{6/(r+c)})$ [12], where r and c are the numbers of rows and columns of the matrixes, respectively. We assign the initial values of **E** with pre-trained word embeddings from a large scale reviews corpus. We set the widths of three convolutional filter as 1, 2 and 3, output length of convolutional filter as 50. The learning rate is set as 0.01. Note that we learn empirically different learning rate, and find that the best performance can be obtained when the learning rate is 0.01.

4. Experiments

4.1. Experimental setup

We use the dataset of Li et al. (2014), which consists of truthful and deceptive reviews in three domains [17], namely *Hotel, Restaurant* and *Doctor*. For each domain, a set of *Customer* reviews are collected as truthful reviews, and a set of deceptive reviews are collected from *Turker* and *Employee*, respectively. We follow Li et al. (2014) in designing the evaluation metrics [17]. For the *Hotel* domain, we perform both three-way (*Customer/Employee/Turker*) and two-way classification between *Customer* reviews and *Employee/Turker* reviews. This is because deceptive reviews from *Employee* and *Turker* can reflect different

Table 1
Statistics dataset.

Domain	Turker	Employee	Customer
Hotel	800	280	800
Restaurant	200	120	400
Doctor	200	32	200

Table 2
Development results.

Method	Accuracy	Macro-F1
Average	73.0	73.9
CNN	75.9	77.4
RNN	63.2	64.8
GRNN	79.0	79.9
Average GRNN	80.1	80.7
Bi-directional average GRNN	**83.6**	**83.4**
Le and Mikolov [2014]	76.1	77.6

levels of domain knowledge. For the *Restaurant* and *Doctor* domains, we perform only two-way *Customer*/*Turker* classification because *Employee* reviews are relatively too few. Table 1 shows the statistics of the dataset. For each experiment, we measure both the per-instance accuracy and the macro-F1 score across different classes.

4.1.1. Word embeddings

we learned word embeddings of 50,100 and 200 dimensions using the CBOW model [29] from a large-scale Amazon reviews corpus[1], which contains 34,686,770 reviews on Amazon products from June 1995 to March 2013. The vocabulary size is about 1M. We find that the best performance can be obtained when the dimension is set as 100. During training, we use the average of all the pre-trained embeddings vectors to initialize a vector for unknown words.

4.2. Development experiments

To compare the effectiveness of various neural document models, we conduct a set of development experiments using the mixed dataset of all three domains. Only *Turker* and *Customer* reviews are used, and the total of 2600 reviews are split randomly into training/tuning/testing sets with a ratio of 80/10/10. The tuning set is used for optimizing the hyper-parameters for each neural network structure. The following neural network structures are compared.

- **Average**: simply using the average of all sentence vectors as the document vector.
- **CNN**: a narrow convolutional neural network with width 3 is used, before the resulting vector of each convolutional function is averaged.
- **RNN**: a standard (single-directional) recurrent neural network is used, with its last state vector being used as the document vector.
- **GRNN**: the (single-directional) gated recurrent neural network of this paper, with its last state vector being used as the document vector.
- **Average GRNN**: the (single-directional) gated recurrent neural network of this paper, with the average of its state vectors being used as the document vector.
- **Bi-directional Average GRNN**: the bi-directional GRNN model of this paper.

Table 2 shows the results. Without modeling discourse relations, the averaging method gives a baseline accuracy of 73.0%. CNN gives better results by capturing relationships between local sentences. Though modeling global sequential relations, RNN does not give better results compared with the averaging baseline and the main reason is vanishing gradients in its training. By using gates, the results of GRNN is significantly better than both the baseline and the CNN document model. Both averaging and the bi-directional extension further increased the accuracies, and our best development result is 83.6%.

We also compare our methods with the paragraph vector model [16], which builds document representation without considering sentence vectors. It gives results comparable to the CNN model, but much lower compared with the GRNN models, which leverage non-local discourse structures.

4.2.1. Influences of word representation

We compare the effect of word embedding initialization method using the development test data. The results are shown in Fig. 4, where pre-trained word embeddings give higher accuracies compared with random initialization with fine-tuning,

[1] http://snap.stanford.edu/data/web-Amazon.html.

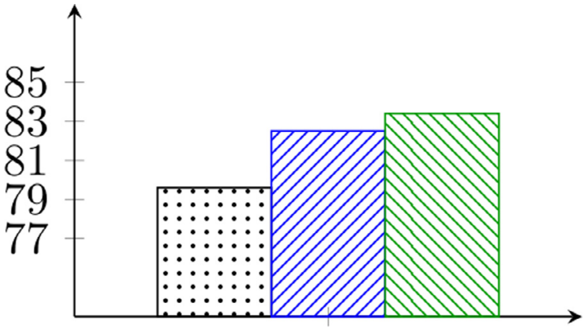

Fig. 4. Influence of word representations, black—random initialization, blue—pre-trained word embeddings, and green—pre-trained word embeddings with fine-tuning. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
In-domain results, ALL represents the three-way classification.

Domain	Setting	Methods	Accuracy	Macro-F1
Hotel	ALL	Li et al.	66.4	67.3
		Neural/Logistic	**78.3**/65.9	**74.0**/66.7
		Integrated	**80.8**	**76.9**
	Customer/Turker	Li et al.	81.8	82.6
		Neural/Logistic	**83.5**/81.9	**83.5**/82.8
		Integrated	**85.7**	**85.6**
	Customer/Employee	Li et al.	79.9	80.9
		Neural/Logistic	**84.3**/79.1	**81.9**/80.2
		Integrated	**87.0**	**84.4**
	Employee/Turker	Li et al.	76.2	78.0
		Neural/Logistic	**90.8**/75.7	**87.5**/77.9
		Integrated	**92.6**	**90.1**
Restaurant	Customer/Turker	Li et al.	81.7	82.2
		Neural/Logistic	**84.4**/82.1	**84.6**/82.3
		Integrated	**86.9**	**86.8**
Doctor	Customer/Turker	Li et al.	74.5	73.5
		Neural/Logistic	74.6/73.7	72.8/72.4
		Integrated	**76.0**	**74.1**

and fine-tuning pre-trained embeddings gives the best results. The findings are consistent with similar investigations for other NLP tasks [5,37,42].

4.3. In-domain results

We choose the best neural model, namely the bi-directional average GRNN, according to the development test results. A set of in-domain test are conducted according to Li et al. (2014)'s settings [17], in order to compare the neural model with state-of-the-art discrete model with SVM. In particular, all results are reported by using ten-fold cross-validation. As mentioned in the introduction, Li et al. (2014) use hand-crafted features that contain word, POS and other linguistic clues [17].

The results are shown in Table 3, in the Li et al. rows and the left items of the Neural/Logistic rows, respectively. For the *Hotel* domain, the neural model outperforms the discrete model of Li et al. (2014) on both three-way *Customer/Employee/Turker* classification and two-way classification tasks. While Li et al. (2014)'s method gives about 80% accuracies on *Customer/Turker* and *Customer/Employee* classifications, which distinguish truthful and deceptive reviews. The accuracies drop to below 66.4% when all three classes are involved. In contrast, our method gives a accuracy of 78.3% for the three-way task, demonstrating the power of the neural model in distinguishing deceptive reviews from different authors. Contrast on the two-way *Employee/Turker* classification task is consistent. This shows the power of the neural model in capturing subtle semantic features which are difficult to express using manual indicator features. Meanwhile, our neural model has advantages over discrete models in computational costs, because the neural model takes distributed word embedding as inputs, and any data preprocessing or cleaning steps are not involved. However, for discrete models, the feature engineering is necessary and very time-consuming. The above analysis show that the neural model is more suitable for deceptive opinion spam detection.

The results on the *Restaurant* domain is similar to those on the *Hotel* domain, where the neural model significantly outperforms the discrete model. However, the neural model gives similar results compared with the discrete model on the

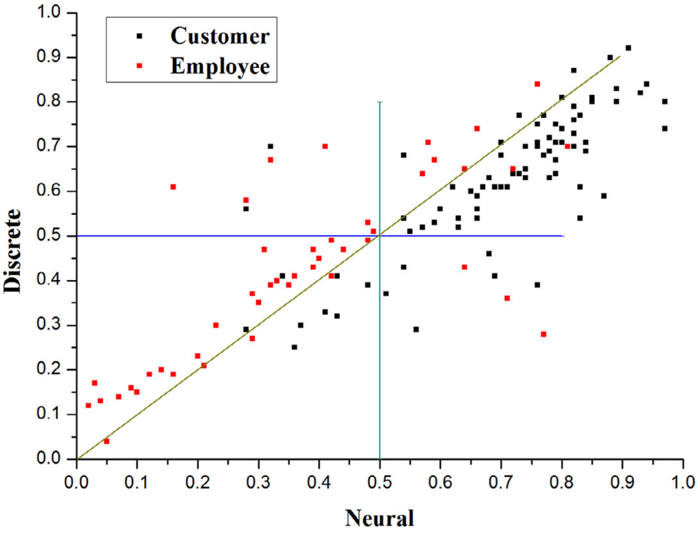

Fig. 5. Output probability comparisons for Hotel (Customer/Employee).

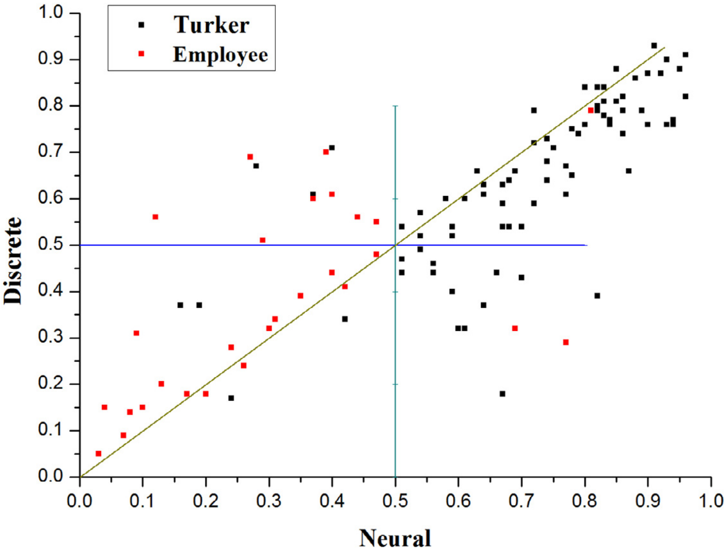

Fig. 6. Output probability comparisons for Hotel (Turker/Employee).

Doctor domain. One possible reason is that that number of reviews in this dataset is relatively lower, which lead to relatively lower accuracies by both models. Another reason of the low results of the neural model is a relatively high OOV rate, and 7.02% of the test words in the *Doctor* domain are out of the embedding dictionary (in contrast to 3.25% in the *Hotel* domain and 3.43% in the *Restaurant* domain).

4.3.1. *Analysis*

In order to contrast the effect on discrete and neural features, we build a discrete model using logistic regression with the same discrete feature as Li et al. (2014). The main advantage of using this model is a direct comparison on features, because a logistic regression classifier is the same as the *softmax* output layer of our neural network model in mathematic form. The only difference is that the logistic regression method uses discrete features, while the neural model uses continuous features from the deep neural network. The results of the logistic regression model are shown in the right items of the Neural/Logistic rows in Table 3, which are slightly lower but comparable to Li et al. (2014)'s SVM results.

Figs. 5 and 6 show the output probabilities of the *Customer* and *Turker* classes by both the neural and the logistic discrete models, respectively. Results on the *Hotel* (*Customer*/*Employee*) and *Hotel* (*Turker*/*Employee*) datasets are shown in Figs. 5 and 6, respectively. The x-axis shows the probability by the neural model and the y-axis shows the probability by the discrete model. Taking Fig. 5 for example, true *Customer* reviews in the test set are shown in black, where false reviews by *Employee* are shown in red. As a result, black dots on the top half of the figure and red dots on the bottom show that the discrete

Table 4
In-domain results with attention mechanism.

Domain	Setting	Methods	Accuracy	Macro-F1
Hotel	ALL	Neural	78.9	74.7
		Integrated	81.3	77.4
	Customer/Turker	Neural	84.1	84.2
		Integrated	86.1	86.0
	Customer/Employee	Neural	84.8	82.4
		Integrated	87.2	84.7
	Employee/Turker	Neural	91.1	87.9
		Integrated	92.8	90.4
Restaurant	Customer/Turker	Neural	84.8	85.0
		Integrated	87.1	87.0
Doctor	Customer/Turker	Neural	75.3	73.4
		Integrated	76.3	74.5

model predicted correctly, while black dots on the right and red dots on the left show that the neural model predicted correctly.

As shown in the Fig. 5, most black dots are on the top-right of the figure and most red dots are on the bottom-left, showing that both models are correct in most cases. However, the dots are relatively more disperse in the x-axis, showing that the neural model is more confident in scoring the inputs. This demonstrates the effectiveness of neural features. Observation in Fig. 6 is similar. For the more challenging task, the neural model shows large advantages. Figs. 5 and 6 also shows that the errors by using neural and discrete features can be complementary, which suggests that integrating both types of features in a single model can further improve the performance. We make a feature integration by directly concatenating the discrete feature vector to the neural features vector before the *softmax* layer. The results of the combined model are shown in the Integrated rows in Table 3. In all the test sets, the model gives significantly[2] better results compared with both the neural and logistic models.

4.3.2. More results

In our work, we apply the GRNN structure over sentence representation vectors in the left-to-right and right-to-left directions, respectively, resulting in a forward state sequence h_1, h_2, .., h_n and a backward state sequence $h^{'}_n, h^{'}_{n-1}, .., h^{'}_1$, respectively. Then a combination of h_i and $h^{'}_i$ is used as its bi-directional state vector for each sentence vector node s_i. Here, all bi-directional state vectors are treated equally, so the noisy or irrelevant part may degrade the classification performance. Meanwhile, Vrij et al. (2009) and Ott et al. (2011) find that different topics have different importance in deceptive opinion detection [34,51]. For example, spatial information can usually be a strong indicator of non-spam for hotel reviews. So we introduce a simple attention mechanism to consider the importance of different state vectors. Specifically, for each sentence s_i in one document d, which contains the sentences vectors s_1, s_2, .., s_m, we integrate the weights into bi-directional state vector h_i and $h^{'}_i$. Specifically, we use the context vector to measure the importance of the sentences. This yields

$$u_i = tanh(W_s(h_i \oplus h^{'}_i) + b_s), \tag{13}$$

$$\beta_i = \frac{exp({u_i}^{\mathrm{T}} u_s)}{\sum_i exp({u_i}^{\mathrm{T}} u_s)} \tag{14}$$

The document vector d is represented as

$$d = \sum_i \beta_i (h_i \oplus h^{'}_i), \tag{15}$$

where $\sum_{i=1}^{m} \beta_i = 1$, and $\oplus$ is the vector concatenation function. The context vector u_s has been used in previous memory networks [27,48], and it can been randomly initialized and jointly learned during the training process.

Table 4 shows the results of our model with attention mechanism. Compared with the results of Table 3, the model gives better performance by introducing the attention mechanism into the bi-directional GRNN. In future work, we will introduce the attention mechanism for the composition of words in forming sentences.

4.4. Cross-domain results

For the task of deceptive opinion spam detection, the sample numbers of dataset is relatively small, and the collection of labeled data is time-consuming and expensive. We investigate two important questions. First, it is interesting to know

[2] The p-value is below 10^{-3} using *t*-test

Table 5
Cross-domain results.

Domain	Methods	Accuracy	Macro-F1
Restaurant	Li et al.	78.5	77.8
	Neural	81.7	80.6
	Integrated	**83.5**	**82.3**
Doctor	Li et al.	55.0	61.7
	Neural	55.7	65.9
	Integrated	**57.0**	**67.4**

whether the relatively more richly annotated *Hotel* domain dataset can be used to train effective deception detection models on the *Restaurant* or *Doctor* domain. Second, we want to study the generalization ability of our neural model. We frame the problems as a domain adaptation task, training a classifier on *Hotel* reviews, and evaluate the performance on the other domains. For simplicity, we focus on two-way *Customer*/*Turker* classification.

The results are shown in Table 5. First, the classifiers trained on *Hotel* reviews apply well in the *Restaurant* domain, which is reasonable due to the many shared properties among *Restaurant* and *Hotel*, such as the environment and location. However, the performance on the *Doctor* domain is much worse, largely due to the difference in vocabulary. Second, compared with the method of Li et al. (2014), our neural model gives better performance. For the *Doctor* domain, both models trained on the *Hotel* domain do not generalize well. Our neural model gives a higher F1 (65.9%) compared with the SVM classifier (61.7%), which shows some relative effectiveness of neural model. Similar to the in-domain results, the integrated model outperforms both the discrete and neural models.

5. Conclusion

We empirically explore a gated recurrent neural network model for deceptive opinion spam detection. For in-domain experiments, because the neural model can properly capture non-local discourse information over sentence vectors, and the neural model outperforms a state-of-the-art discrete baseline, and also simple neural document models such as paragraph vectors. Then, experimental results show that the neural model with attention mechanism outperforms the model without attention mechanism. For cross-domain experiments, the results show that the neural model gives a higher performance than the discrete model. This shows that the neural model has stronger generalization ability compared with discrete model. Besides, further experiments show that the accuracies can be improved by integrating discrete and neural features.

Acknowledgments

This work is supported by the State Key Program of National Natural Science Foundation of China (Grant No.61133012), the National Natural Science Foundation of China (Grant No.61173062, 61373108) and the National Philosophy Social Science Major Bidding Project of China (Grant No. 11&ZD189).

References

[1] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Networks 5 (2) (1994) 157–166.
[2] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. (3) (2003) 1137–1155.
[3] K. Cho, B.V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).
[4] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Gated feedback recurrent neural networks, arXiv preprint arXiv:1502.02367 (2015).
[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (2011) 2493–2537.
[6] N.F.F.d. Silva, L.F. Coletta, E.R. Hruschka, E.R. Hruschka Jr, Using unsupervised information to improve semi-supervised tweet sentiment classification, Inf. Sci. (Ny) 355 (2016) 348–365.
[7] L. Dong, F. Wei, M. Zhou, K. Xu, Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1537–1543.
[8] H. Drucker, D. Wu, V.N. Vapnik, Support vector machines for spam categorization, IEEE Trans. Neural Netw. 10 (5) (1999) 1048–1054.
[9] S. Feng, R. Banerjee, Y. Choi, Syntactic stylometry for deception detection, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2012, pp. 171–175.
[10] V.W. Feng, G. Hirst, Detecting deceptive opinions with profile compatibility, in: Proceedings of the 6th International Joint Conference on Natural Language Processing, 2013, pp. 338–346.
[11] F. Figueiredo, J.M. Almeida, M.A. Gonalves, F. Benevenuto, Trendlearner: early prediction of popularity trends of user generated content, Inf. Sci. (Ny) 349 (2016) 172–187.
[12] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: A deep learning approach, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 513–520.
[13] K.M. Hermann, P. Blunsom, The role of syntax in vector space models of compositional semantics, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2013, pp. 894–904.
[14] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[15] O. Irsoy, C. Cardie, Deep recursive neural networks for compositionality in language, in: Advances in Neural Information Processing Systems, 2014, pp. 2096–2104.
[16] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, arXiv preprint arXiv:1405.4053(2014).

[17] J. Li, M. Ott, C. Cardie, E.H. Hovy, Towards a general rule for identifying deceptive opinion spam, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2014, pp. 1566–1576.
[18] J. Li, Feature weight tuning for recursive neural networks, arXiv preprint arXiv:1412.3714 (2014).
[19] J. Li, D. Jurafsky, E. Hovy, When are tree structures necessary for deep learning of representations?, arXiv preprint arXiv:1503.00185 (2015).
[20] J. Li, M. Luong, D. Jurafsky, A hierarchical neural autoencoder for paragraphs and documents, arXiv preprint arXiv:1506.01057 (2015).
[21] N. Jindal, B. Liu, Opinion spam and analysis, in: Proceedings of the 2008 International Conference on Web Search and Data Mining, ACM, 2008, pp. 219–230.
[22] R. Johnson, T. Zhang, Effective use of word order for text categorization with convolutional neural networks, arXiv preprint arXiv:1412.1058 (2014).
[23] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, arXiv preprint arXiv:1404.2188 (2014).
[24] S. Kc, A. Mukherjee, On the temporal dynamics of opinion spamming: case studies on yelp, in: Proceedings of the 25th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 369–379.
[25] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014).
[26] S. Kim, H. Chang, S. Lee, M. Yu, J. Kang, Deep semantic frame-based deceptive opinion spam analysis, in: Proceedings of the ACM International Conference on Information and Knowledge Management, 2015, pp. 1131–1140.
[27] A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, R. Socher, Ask me anything: dynamic memory networks for natural language processing, arXiv preprint arXiv:1506.07285 (2015).
[28] C. Miller, Company settles case of reviews it faked, new york times.
[29] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[30] S.M. Mohammad, S. Kiritchenko, X. Zhu, Nrc-canada: building the state-of-the-art in sentiment analysis of tweets, Comput. Sci. (2013).
[31] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, R. Ghosh, Spotting opinion spammers using behavioral footprints, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 632–640.
[32] M.L. Newman, J.W. Pennebaker, D.S. Berry, J.M. Richards, Lying words: predicting deception from linguistic styles, Person. Soc. Psychol. Bull. 29 (5) (2003) 665–675.
[33] A. Ntoulas, M. Najork, M. Manasse, D. Fetterly, Detecting spam web pages through content analysis, in: Proceedings of the 15th international conference on World Wide Web, ACM, 2006, pp. 83–92.
[34] M. Ott, Y. Choi, C. Cardie, J.T. Hancock, Finding deceptive opinion spam by any stretch of the imagination, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, pp. 309–319.
[35] M. Ott, C. Cardie, J. Hancock, Estimating the prevalence of deception in online review communities, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 201–210.
[36] R. Paulus, R. Socher, C.D. Manning, Global belief recursive neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 2888–2896.
[37] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2014, pp. 1532–1543.
[38] T. Qian, B. Liu, Identifying multiple userids of the same author, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2013, pp. 1124–1135.
[39] Y. Ren, D. Ji, H. Zhang, Positive unlabeled learning for deceptive reviews detection, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2014, pp. 488–498.
[40] Y. Ren, D. Ji, L. Yin, H. Zhang, Finding deceptive opinion spam by correcting the mislabeled instances, Chin. J. Electr. 24 (1) (2015) 52–57.
[41] Y. Ren, R. Wang, D. Ji, A topic-enhanced word embedding for twitter sentiment classification, Inf. Sci. (Ny) 369 (2016) 188–198.
[42] Y. Ren, Y. Zhang, M. Zhang, D. Ji, Context-sensitive twitter sentiment classification using neural network, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 215–221.
[43] Y. Ren, Y. Zhang, M. Zhang, D. Ji, Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 3038–3044.
[44] J.K. Rout, S. Singh, S.K. Jena, S. Bakshi, Deceptive review detection using labeled and unlabeled data, Multimed. Tools Appl. (2016) 1–25.
[45] C.N.D. Santos, M. Gattit, Deep convolutional neural networks for sentiment analysis of short texts, Int. Conf. Comput. Linguist. (2014) 69–78.
[46] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2013, pp. 1631–1642.
[47] D. Streitfeld, For $2 a star, an online retailer gets 5-star product reviews, new york times.
[48] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, Weakly supervised memory networks, arXiv preprint arXiv:1503.08895 (2015).
[49] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for twitter sentiment classification, in: Meeting of the Association for Computational Linguistics, 2014, pp. 1555–1565.
[50] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 1422–1432.
[51] A. Vrij, S. Leal, e.a. Granhag, Outsmarting the liars: the benefit of asking unanticipated questions, Law Hum. Behav. 33 (2) (2009) 159–166.
[52] D. Wang, E. Nyberg, A long short-term memory model for answer sentence selection in question answering, in: Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing, 2015, pp. 707–712.
[53] A. Yessenalina, C. Cardie, Compositional matrix-space models for sentiment analysis, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 172–182.
[54] K.-H. Yoo, U. Gretzel, Comparison of deceptive and truthful travel reviews, Inf. Commun. Technol. Tourism (2009) 37–47.
[55] Z. Gyöngyi, H. Garcia-Molina, J. Pedersen, Combating web spam with trustrank, in: Thirtieth International Conference on Very Large Data Bases, 2004, pp. 576–587.