Deceptive Opinion Spam Detection Using Neural Network

Yafeng Ren and Yue Zhang
Singapore University of Technology and Design, Singapore
renyafeng@whu.edu.cn, yue_zhang@sutd.edu.sg

Abstract

Deceptive opinion spam detection has attracted significant attention from both business and research communities. Existing approaches are based on manual discrete features, which can capture linguistic and psychological cues. However, such features fail to encode the semantic meaning of a document from the discourse perspective, which limits the performance. In this paper, we empirically explore a neural network model to learn document-level representation for detecting deceptive opinion spam. In particular, given a document, the model learns sentence representations with a convolutional neural network, which are combined using a gated recurrent neural network with attention mechanism to model discourse information and yield a document vector. Finally, the document representation is used directly as features to identify deceptive opinion spam. Experimental results on three domains (*Hotel*, *Restaurant*, and *Doctor*) show that our proposed method outperforms state-of-the-art methods.

1 Introduction

Online reviews on products and services are extensively used by consumers and businesses for conducting decisive purchase, making product design and altering marketing strategies. As a result, deceptive opinion spam (e.g. deceptive reviews) arouses increasing attention (Streitfeld, 2012). Opinion spam is a type of review with fictitious opinions, deliberately written to sound authentic (Jindal and Liu, 2008; Ott et al., 2011). It can be difficult for human readers to distinguish them from truthful reviews. In a test by Ott et al. (2011), the average accuracy of three human judges is only 57.33%. It can be expensive to detect opinion spam manually over large user-generated texts. Hence, machine learning methods for automatically detecting deceptive opinion spam can be useful.

The objective of the task is to identify whether a given document is a spam or not. The majority of existing approaches follow the seminal work of Jindal and Liu (2008), employing classifiers with supervised learning. Most studies focus on designing effective features to enhance the 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) (Newman et al., 2003) features. 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 (Le and Mikolov, 2014; Tang et al., 2015), achieving highly competitive results. Potential advantages of using neural networks for spam detection are three-fold. First, neural models use dense 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 a large-scale raw text, thus alleviating the sparsity of annotated data to some

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 140–150, Osaka, Japan, December 11-17 2016.

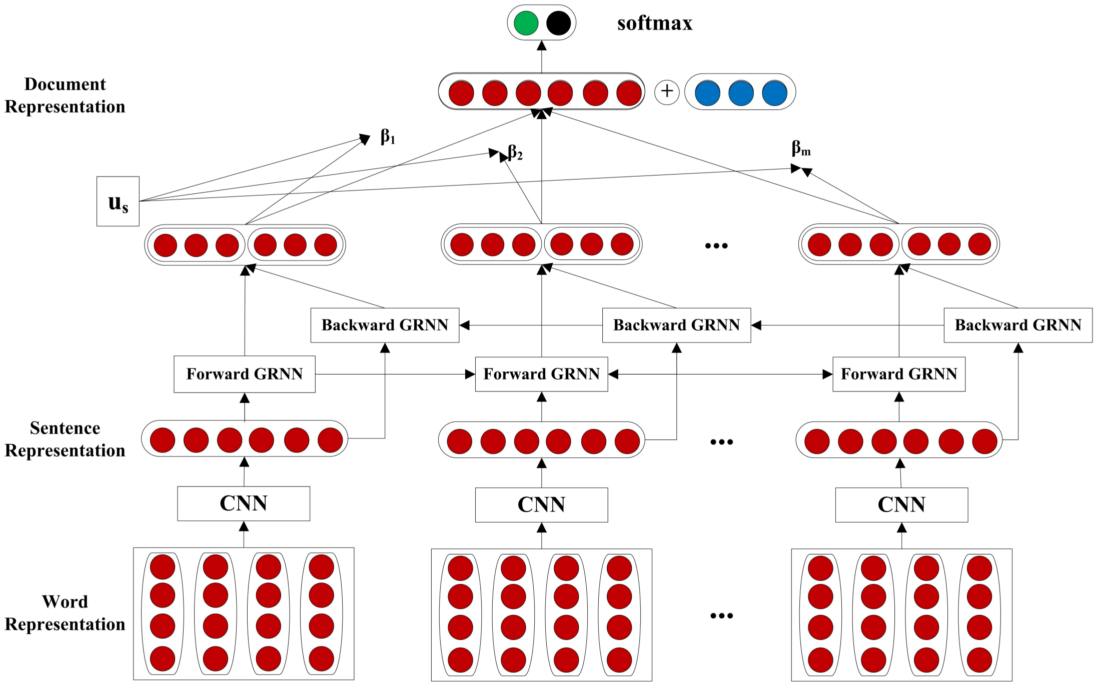


Figure 1: Neural network model structure for deceptive opinion spam detection, red nodes represent neural features, and blue nodes represent discrete features.

extent. Third, neural network models can be used to induce document representations from sentence representations, leveraging sentence and discourse information.

In this paper, we empirically investigate the effectiveness of learning dense document representations for opinion spam detection. In particular, we propose a three-stage neural network system, as shown in Figure 1. In the first stage, a convolutional neural network is used to produce sentence representations from word representations. Second, a bi-directional gated recurrent neural network with attention mechanism is used to construct a document representation from the sentence vectors. 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 compare the proposed models on a standard benchmark (Li et al., 2014), which consists of data from three domains (*Hotel*, *Restaurant*, and *Doctor*). Results on in-domain and cross-domain experiments show that the dense neural features significantly outperforms the previous state-of-the-art methods, demonstrating the advantage of neural models in capturing semantic characteristics. In addition, automatic neural features and manual discrete features are complementary sources of information, and a combination leads to further improvements.

2 Related Work

2.1 Deceptive Opinion Spam Detection

Spam detection has been extensively investigated in the Web-page and E-mail domains (Gyöngyi et al., 2004; Ntoulas et al., 2006), while research has recently been extended to the customer review domain (Ott et al., 2011; Mukherjee et al., 2013; Li et al., 2014). Various types of indicator features have been investigated. For examples, Jindal and Liu (2008) trained models using features based on the review content, the reviewer, and the product itself. Yoo and Gretzel (2009) gathered 40 truthful and 42 deceptive hotel reviews and manually compared the linguistic differences between them.

Ott et al. (2011) created a benchmark dataset by employing *Turkers* to write fake reviews. Their data were adopted by a line of subsequent work (Ott et al., 2012; Feng et al., 2012; Feng and Hirst, 2013). For example, Feng et al. (2012) looked into syntactic features from Context Free Grammar (CFG) parse trees to improve the performance. 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. Recently, Li et al. (2014) created a wider-coverage benchmark, 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. To our knowledge, we are the first to investigate deep learning for deceptive opinion spam detection.

There has been work that exploits features outside the review content itself. In addition to Jindal and Liu (2008), Mukherjee et al. (2013) explored the features from customer's behavior to identify deception. Based on some truthful reviews and a lot of unlabeled reviews, Ren et al. (2014) proposed a semi-supervised learning method, and built an accurate classifier to identify deceptive reviews. Kim et al. (2015) introduced a frame-based semantic feature based on FrameNet. Experimental results show that semantic frame features can improve the classification accuracy. We focus on the review content in this paper, but their features can be used to extend our model.

2.2 Neural Network Models for Representation Learning

Neural network models have been exploited to learn dense feature representation for a variety of NLP tasks (Collobert et al., 2011; Kalchbrenner et al., 2014; Ren et al., 2016b). Distributed word representations (Mikolov et al., 2013) 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, Le and Mikolov (2014) introduced paragraph vector to learn document representations, extending to word embedding methods of Mikolov et al. (2013). Socher et al. (2013) introduced a family of recursive neural networks to represent sentence-level semantic composition. Follow-up research includes recursive neural network with global feed backward mechanisms (Paulus et al., 2014), deep recursive layers (Irsoy and Cardie, 2014), and adaptive composition functions (Dong et al., 2014).

Convolutional neural networks have been widely used for semantic composition (Kalchbrenner et al., 2014; Johnson and Zhang, 2014), automatically capturing n-gram information. Sequential models such as recurrent neural network or long short-term memory (LSTM) (Li et al., 2015a; Tang et al., 2015) have also been used for recurrent semantic composition. The attention mechanism was first proposed in machine translation (Bahdanau et al., 2014). Further uses of the attention mechanism include parsing (Vinyals et al., 2014), natural language question answering (Sukhbaatar et al., 2015; Kumar et al., 2015; Hermann et al., 2015), and image question answering (Yang et al., 2015). We explore CNN and recurrent neural networks with attention mechanism to learn document representation for detecting deceptive opinion spam, comparing their effect with bag-of-word and paragraph vector baselines.

3 Approach

The proposed neural network model learns real-valued dense vector representations for documents of variable lengths, which is used as the feature to classify each document. Shown in Figure 1, it consists of two main components, The first produces distributed vector sentence representations from word representations, and the second gives dense vector document representations from the sentence vectors.

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 (Johnson and Zhang, 2014; Ren et al., 2016a), while LSTM gives the best results for question answering (Wang and Nyberg, 2015). For modeling discourse structures, LSTM has been used more frequently (Li et al., 2015b; Tang et al., 2015). 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 Figure 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 words using embeddings (Bengio et al., 2003), which are low-dimensional dense real-valued vectors. For each word w, we use a look-up matrix E to obtain its embedding $e(w) \in R^D$, where $E \in R^{D \times V}$ is a model parameter, D is the word vector dimension size and V is the vocabulary size. E can be randomly initialized from a uniform distribution (Socher et al., 2013), or pre-trained from a large raw corpus (Mikolov et al., 2013).

As shown in the bottom of Figure 1, a convolutional neural network (CNN) (Kim, 2014; Kalchbrenner et al., 2014; Johnson and Zhang, 2014) is used to learn dense representations of a sentence. We use three convolutional filters to capture the local semantics of n-grams of various granularities. 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. We set $D_1 = 1$, $D_2 = 2$ and $D_3 = 3$ for representing unigrams, bigrams and trigrams, respectively. Taking D_2 for example, W_2 and b_2 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_2, which is denoted as $I_{2,i} = [e(w_i); e(w_{i+1}); ...; e(w_{i+D_2-1})] \in R^{D \times D_2}$. The output of a linear layer is calculated as

$$H_{2,i} = W_2 \cdot I_{2,i} + b_2, \tag{1}$$

where $W_2 \in R^{l_{oc} \times D \times D_2}$, l_{oc} is the output size of the linear layer. We use an average pooling layer to merge the varying number of outputs $\{H_{2,1}, H_{2,2}, .., H_{2,n}\}$ from the convolution layer into a vector with fixed dimensions.

$$H_2 = \frac{1}{n} \sum_{i=1}^{n} H_{2,i} \tag{2}$$

To incorporate nonlinearity, a activation function $tanh$ is used to obtain the output O_2 of this filter.

$$O_2 = tanh(H_2) \tag{3}$$

Similarly, we obtain the O_1 and O_3 for the other two convolutional filters with width 1 and 3, respectively. The outputs of three filters are lastly averaged to generate sentence representation.

3.2 Document Model

Given a document with m sentences, we use the sentence vectors $s_1, s_2, .., s_m$ obtained by the CNN model as inputs, and learn document composition with a gated recurrent neural network (GRNN). Standard recurrent neural networks (RNN) map sentence vectors of variable lengths to a fixed-length vector, by starting with an 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*

$$h_t = tanh(W_r \cdot [h_{t-1}; s_t] + b_r), \tag{4}$$

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 (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997). This makes it difficult to model long-distance correlation in a sequence. We explore a gated recurrent neural network (GRNN) to address this, which is similar in spirit to LSTM (Cho et al., 2014; Chung et al., 2015), but empirically runs faster. Specifically, the transition function of the GRNN used in the work is calculated as follows

$$i_t = sigmoid(W_i \cdot [h_{t-1}; s_t] + b_i) \tag{5}$$

$$f_t = sigmoid(W_f \cdot [h_{t-1}; s_t] + b_f) \tag{6}$$

$$g_t = tanh(W_r \cdot [h_{t-1}; s_t] + b_r) \tag{7}$$

$$h_t = tanh(i_t \odot g_t + f_t \odot h_{t-1}) \tag{8}$$

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.

To better 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. Here, if all bi-directional state vectors are treated equally, 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. 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

$$u_i = tanh(W_s(h_i \oplus h^{'}_i) + b_s), \tag{9}$$

$$\beta_i = \frac{exp({u_i}^{\mathrm{T}} u_s)}{\sum_i exp({u_i}^{\mathrm{T}} u_s)} \tag{10}$$

The document vector d is represented as

$$d = \sum_i \beta_i (h_i \oplus h^{'}_i), \tag{11}$$

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 (Kumar et al., 2015; Sukhbaatar et al., 2015), and it can been randomly initialized and jointly learned during the training process.

3.3 The Classification Model

We use the document representation as features for identifying deceptive opinion spam. Specifically, a linear layer is added to transform the document vector into a real-valued vector, whose length is class number *C*. A *softmax* function is added to convert real vector to conditional probability for document classification.

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,

$$L(\theta) = -\sum_{i=1}^{N} \log \frac{e^{\vec{o}(y_i)}}{e^{\vec{o}(0)} + e^{\vec{o}(1)}} + \frac{\lambda}{2} \parallel \theta \parallel^2, \tag{12}$$

where θ is the set of model parameters.

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

$$\theta_{t,i} = \theta_{t-1,i} - \frac{\alpha}{\sqrt{\sum_{t'=1}^{t} g_{t',i}^2}} g_{t,i}, \tag{13}$$

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

We initialize all the matrix and vector parameters with uniform samples in $(-\sqrt{6/(r+c)}, -\sqrt{6/(r+c)})$, where r and c are the numbers of rows and columns of the matrixes, respectively. We learn word embeddings of 100-dimensions using the CBOW model of Mikolov

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

Table 1: Statistics dataset.

Method	Accuracy (%)	Macro-F1 (%)
Average	73.0	73.9
CNN	75.9	77.4
RNN	63.2	64.8
GRNN	80.1	80.7
Bi-directional GRNN	83.6	83.4
Bi-directional GRNN (Attention)	**84.1**	**83.9**
Le and Mikolov (2014)	76.1	77.6

Table 2: Development results.

et al. (2013) from a large-scale Amazon reviews corpus [1]. During training, we use the average of all the pre-trained embeddings vectors to initialize unknown words. We set the output length of the convolutional filter as 50. The initial learning rate of Adagrad is set as 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, 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 *Turkers* and *Employees*, respectively. We follow Li et al. (2014) in designing the evaluation metrics. 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 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.

4.2 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.

We compare a set of methods for document modeling, which include a single averaging method, tracking a document as a bag of sentences (Average), a CNN, a naive RNN, and our gated RNN in single- and bi-directional method. In addition, we compare the bi-directional RNN without attention and with attention being used.

Table 2 show 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. By introducing the attention mechanism into the bi-directional GRNN, the best development result is 84.1%.

We also compare our methods with the paragraph vector model of Le and Mikolov (2014), which builds a document representation without considering sentence vectors. It gives results comparable to

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

Domain	Setting	Method	Accuracy (%)	Macro-F1 (%)
Hotel	Customer/Employee/Turker	Li et al.	66.4	67.3
		Neural/Logistic	**78.9**/66.5	**74.7**/67.6
		Integrated	**81.3**	**77.4**
	Customer/Turker	Li et al.	81.8	82.6
		Neural/Logistic	**84.1**/82.4	**84.2**/83.5
		Integrated	**86.1**	**86.0**
	Customer/Employee	Li et al.	79.9	80.9
		Neural/Logistic	**84.8**/79.4	**82.4**/80.6
		Integrated	**87.2**	**84.7**
	Employee/Turker	Li et al.	76.2	78.0
		Neural/Logistic	**91.1**/76.2	**87.9**/78.5
		Integrated	**92.8**	**90.4**
Restaurant	Customer/Turker	Li et al.	81.7	82.2
		Neural/Logistic	**84.8**/82.5	**85.0**/82.7
		Integrated	**87.1**	**87.0**
Doctor	Customer/Turker	Li et al.	74.5	73.5
		Neural/Logistic	75.3/74.4	73.4/72.9
		Integrated	**76.3**	**74.5**

Table 3: In-domain results.

the CNN model, but much lower compared with the GRNN models, which leverage non-local discourse structures.

4.3 In-Domain Results

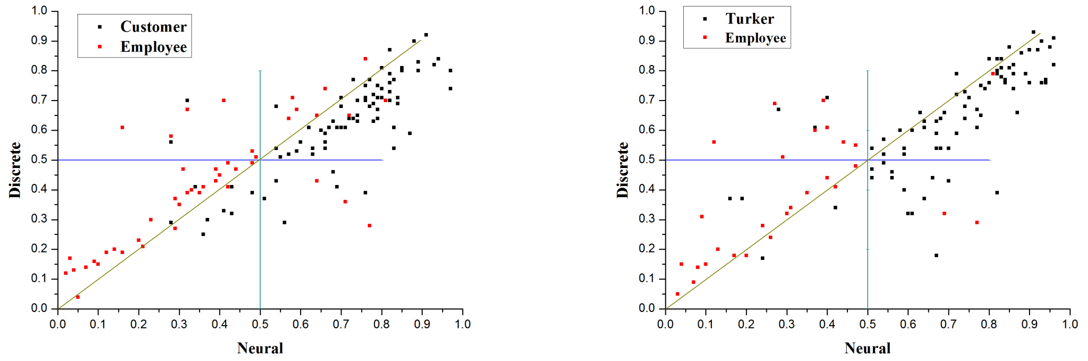
We choose the best neural model, namely the bi-directional GRNN (Attention), according to the development test results. A set of in-domain test are conducted according to Li et al. (2014)'s settings, in order to compare the neural model with the state-of-the-art discrete model with SVM. In particular, 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 the word, POS and other linguistic clues.

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 around 80% accuracies on *Customer/Turker* and *Customer/Employee* classifications, which distinguish truthful and deceptive reviews. The accuracies drop to below 66.4% when all the three classes are involved. In contrast, our method gives an accuracy of 78.9% for the three-way task, demonstrating the power of the neural model in distinguishing deceptive reviews from different types of authors. The 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.

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 *Doctor* domain. One possible reason is that number of reviews in this dataset is relatively lower, which leads to relatively lower accuracies by both models. The other reason 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.



(a) Hotel (Customer/Employee) (b) Hotel (Turker/Employee)

Figure 2: Output probability comparisons.

Figure 2 shows 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 Figure 2(a) and 2(b), respectively. The x-axis shows the probability by the neural model and the y-axis shows the probability by the discrete model. Taking Figure 2(a) 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 of the figure and red dots on the bottom show cases which the discrete model predicted correctly, while black dots on the right and red dots on the left show cases which the neural model predicted correctly.

As shown in the figure, 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 Figure 2(b) is similar. For the more challenging task, the neural model shows large advantages.

Figure 2 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 results. We make a feature integration by directly concatenating the discrete feature vector (the blue nodes in Figure 1) 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 better results compared with both the neural and logistic models[2].

4.4 Cross-Domain Results

For the task of deceptive opinion spam detection, the sample numbers of the dataset are relatively small, and the collection of labeled data is time-consuming and expensive. We investigate two important questions. First, it is interesting to know 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 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 4. First, the classifiers trained on *Hotel* reviews apply well to 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 (66.3%) compared with the SVM classifier

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

Domain	Method	Accuracy (%)	Macro-F1 (%)
Restaurant	Li et al.	78.5	77.8
	Neural	81.9	81.0
	Integrated	**83.7**	**82.6**
Doctor	Li et al.	55.0	61.7
	Neural	56.1	66.3
	Integrated	**57.3**	**67.6**

Table 4: Cross-domain results.

(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 investigated a gated recurrent neural network model with attention mechanism for deceptive opinion spam detection. By capturing non-local discourse information over sentence vectors, the neural network model outperforms a state-of-the-art discrete baseline, and also simple neural document models such as paragraph vectors. Further experiments show that the accuracies can be improved by integrating discrete and neural features.

Acknowledgements

We thank all reviewers for all their detailed comments. This work supported by Singapore Ministry of Eduction (MOE) AcRF Tier 2 grant T2MOE201301. We also thank Jiayuan Deng for some experimental work in the early stage of this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. *arXiv preprint arXiv:1502.02367*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2014. Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Vanessa Wei Feng and Graeme Hirst. 2013. Detecting deceptive opinions with profile compatibility. In *Proceedings of the International Joint Conference on Natural Language Processing*.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. 2004. Combating web spam with trustrank. In *Proceedings of the International Conference on Very Large Data Bases*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *arXiv preprint arXiv:1506.03340*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*.

Nitin Jindal and Bing Liu. 2008. Opinion spam and analysis. In *Proceedings of the International Conference on Web Search and Web Data Mining*.

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Seongsoon Kim, Hyeokyoon Chang, Seongwoon Lee, Minhwan Yu, and Jaewoo Kang. 2015. Deep semantic frame-based deceptive opinion spam analysis. In *Proceedings of the ACM International Conference on Information and Knowledge Management*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. 2014. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Jiwei Li, Dan Jurafsky, and Eudard Hovy. 2015a. When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185*.

Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. 2015b. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.

Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Spotting opinion spammers using behavioral footprints. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. 2003. Lying words: Predicting deception from linguistic styles. *Personality and Social Psychology Bulletin*.

Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. 2006. Detecting spam web pages through content analysis. In *Proceedings of the International World Wide Web Conference*.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Myle Ott, Claire Cardie, and Jeff Hancock. 2012. Estimating the prevalence of deception in online review communities. In *Proceedings of the International World Wide Web Conference*.

Romain Paulus, Richard Socher, and Christopher D Manning. 2014. Global belief recursive neural networks. In *Advances in Neural Information Processing Systems*.

Yafeng Ren, Donghong Ji, and Hongbin Zhang. 2014. Positive unlabeled learning for deceptive reviews detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016a. Context-sensitive twitter sentiment classification using neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. 2016b. Improving twitter sentiment classification using topic-enriched multi-prototype word embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

David Streitfeld. 2012. For $2 a star, an online retailer gets 5-star product reviews. *New York Times*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. Weakly supervised memory networks. *arXiv preprint arXiv:1503.08895*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. 2014. Grammar as a foreign language. *arXiv preprint arXiv:1412.7449*.

Aldert Vrij, Sharon Leal, and et al. Granhag. 2009. Outsmarting the liars: the benefit of asking unanticipated questions. *Law and human behavior*.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. *Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola. 2015. Stacked attention networks for image question answering. *arXiv preprint arXiv:1511.02274*.

Kyung-Hyan Yoo and Ulrike Gretzel. 2009. Comparison of deceptive and truthful travel reviews. *Information and communication technologies in tourism 2009*.