Computer and Information Science; Vol. 12, No. 2; 2019
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Opinion Spam Detection based on Annotation Extension and Neural Networks

Yuanchao Liu[1] & Bo Pang[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology, China

Correspondence: Yuanchao Liu, School of Computer Science and Technology, Harbin Institute of Technology, China. Tel: 86-0451-8641-3322. E-mail: ycliuharbin@163.com

Received: January 2, 2019 Accepted: March 3, 2019 Online Published: April 5, 2019

doi:10.5539/cis. v12n2p87 URL: https://doi.org/10.5539/cis.v12n2p87

Abstract

Online reviews play an increasingly important role in the purchase decisions of potential customers. Incidentally, driven by the desire to gain profit or publicity, spammers may be hired to write fake reviews and promote or demote the reputation of products or services. Correspondingly, opinion spam detection has attracted attention from both business and research communities in recent years. However, unlike other tasks such as news classification or blog classification, the existing review spam datasets are typically limited due to the expensiveness of human annotation, which may further affect detection performance even if excellent classifiers have been developed. We propose a novel approach in this paper to boost opinion spam detection performance by fully utilizing the existing labelled small-size dataset. We first design an annotation extension scheme that uses extra tree classifiers to train multiple estimators and then iteratively generate reliable labelled samples from unlabeled ones. Subsequently, we examine neural network scenarios on a newly extended dataset to learn the distributed representation. Experimental results suggest that the proposed approach has better generalization capability and improved performance than state-of-the-art methods.

Keywords: opinion spam detection, annotation extension, neural networks

1. Introduction

Product reviews have played an increasingly important role in the purchase decisions of potential customers. A study in Harvard University (Luca, 2013) showed that a one-point increment in star rating can increase revenue by 5%–9%. In consideration of earning financial incentives, spammers may be hired to write fake reviews to promote or demote the reputation of products or services. Genuine reviews are helpful for customers who intend to buy the right product and for businesses that aim to improve product quality. By contrast, fake comments constitute interference or noise, mislead potential customers, and produce adverse effects on the Internet economy.

Many approaches for opinion spam detection are based on standard supervised learning techniques (Ott et al., 2011; Ott et al., 2013). A fake review often starts by praising certain products, belittling others, and inevitably exhibits many external features (Ren et al., 2017). Therefore, researchers have studied such fake reviews by applying supervised models to train the classifiers and predict the unknown reviews (Heydari et al., 2015; Cardoso et al., 2018). The research on text classification is relatively mature and has been proven effective for other problems, such as e-mail spam and blog classification, and many methods can be applied for opinion spam detection.

Opinion spam is more difficult (Li et al., 2015; Santosh et al., 2016) than the many other generic web spam classification tasks, e.g., link spam (Shen et al., 2006), email spam (Chirita et al., 2005), and blog spam (Kolari et al., 2006). One of the challenges in the opinion spam detection task is obtaining the ground-truth datasets. An experienced review spammer writes fake reviews very subtly such that even a human expert may not be able to distinguish between genuine and fake reviews by simply reading the content (Heydari et al., 2015). Correspondingly, a human may find annotating the spamicity of this review difficult by simply reading the same content. Thus, constructing the research datasets by inviting people to write fake/truthful reviews to mimic spammers' writing may be more reasonable. To this end, Ott et al. (2011; 2013) used Amazon Mechanical Turk (AMT) to crowdsource anonymous online workers (called turkers) to write fake hotel reviews and generate one gold standard corpus of deceptive opinion spam (Note 1). Thus, fake review detection can be regarded as a text classification problem, and the results from using only linguistic features are encouraging (Ott et al., 2011; Ott et al., 2013; Li et al., 2017).

However, unlike other spam classification tasks, the existing review spam datasets are typically small in size and scarce due to the expensiveness of recruiting turkers. For example, Ott's datasets have only several hundred spam reviews (Ott et al., 2013). Consequently, simulating complex fake reviews is difficult in real situations and may further affect the spam review detection performance even if excellent classifiers have been developed. On the contrary, although large quantities of unlabeled reviews are readily available in many e-commerce websites, the information cannot be used for standard supervised classification techniques (Piroonsup et al., 2018; Wu et al., 2018).

In this study, we attempt to address the popular task of spam review detection by fully utilizing the existing small-size spam review dataset, which is highly costly to obtain. We propose a semi-supervised ensemble learning-based annotation extension scheme that trains multiple estimators to extend spam review label set with reliable unlabeled samples. We then conduct several experiments by training state-of-the-art neural network models on extended dataset to examine the extension classification performance. Neural networks have been proven highly effective for text classification tasks, and large-size datasets are usually preferable for these models to achieve better performance. To the best of our knowledge, such efforts on employing neural networks on extended spam review datasets have never been seen in any prior work.

The main contributions of this study are listed as follows:

- We propose a semi-supervised self-training based annotation extension scheme that trains multiple classifiers to extend the existing spam review label set from unlabeled samples. The goal of our proposed framework is to fully utilize the existing small-size spam review dataset and improve the performance of supervised algorithms by using abundant unlabeled data.
- We examine several neural network scenarios on the extended and the original dataset to effectively capture the semantics of context and then learn the distributed representation for classification. Neural networks have been proven highly effective for text classification tasks, and large-size datasets are usually preferred for these models to enhance the performance.
- We conduct extensive experiments, including the extension strategy and subsequent training of neural network models on the extended dataset, to evaluate the performance of our approaches. The results show that the method yields state-of-the-art performances, and our proposed approach is appropriate for this task.

The rest of this paper is organized as follows. The related work is surveyed in Section 2. The proposed semi-supervised annotation strategy and the neural models are presented in Sections 3 and 4, respectively. Our evaluation process and the experiment results are demonstrated in Section 5. Finally, conclusions and future directions are drawn in Section 6.

2. Related Work

We present a brief review of the related work mainly from three perspectives, namely, deceptive opinion spam detection, semi-supervised self-labelled techniques, and neural networks for learning the distributed representation.

2.1 Deceptive Opinion Spam Detection

To the best of our knowledge, deceptive opinion spam was first investigated by Jindal et al. (2008). Since then, this problem has attracted significant attention from both business and research communities (Ott et al., 2011; Ott et al., 2012; Fei et al., 2013; Wang et al., 2011; Wang et al., 2016; Savage et al. 2015; and Xu et al., 2013).

Some related research attempted to utilize the filtered results of commercial websites, e.g., Yelp, to form the standard dataset for opinion spam detection research (Luca et al., 2013; Mukherjee et al., 2013a; and Santosh et al., 2016). In their work, the reviews filtered by the websites were considered deceptive (fake/spam), whereas others were considered truthful. Li et al. (2014) also discussed the lack of ground truth data for this task and analyzed the filtering results of another Chinese commercial website, Dianping (the last author from Dianping). They claimed that Dianping's algorithm has a highly high precision, but the recall is difficult to determine. The merit for such kinds of work is that the data are usually abundant, and many rich behavior features can be utilized (Mukherjee et al., 2013a; Rayana et al., 2015). The commercial anti-fraud filter is also conducted by algorithm, and the results may be not perfect. Hence, they are only called “near” ground truth. Obtaining ground truths for opinion detection research may only be possible upon spammer confessions or sting operations (Mukherjee et al., 2015), which cannot be performed at large scale (Santosh et al., 2016).

By comparison, Ott et al. (2011) used AMT to crowdsource anonymous online turkers to construct the text-based spam review dataset; thus, they are real spams; several text-classification-based work have been performed based on this dataset (Feng et al. 2012; Hai et al., 2016). Although expert annotation cannot fully eliminate the possibility of any noise, the spam reviews created by turkers may be more reliable because these reviews are real spams. Due to the expensiveness of recruiting turkers, such kinds of existing datasets are usually limited (Hai et al., 2016). Li et al. (2014) also released another spam review dataset, which has a style similar to that of Ott's dataset, but is still relatively small in scale.

The lack of ground truth datasets also renders performance evaluation in spam review detection difficult. Indirect review classification-based evaluation method is then used as an alternative (Mukherjee et al., 2013a; Xu et al., 2015). The idea is that if the detection model is effective, then the top k% reviews will tend to be spams than the bottom k% reviews. Two different review datasets can then be generated. Supervised text classification using n-gram features is effective in detecting spam reviews (Ott et al., 2011; Ott et al., 2013). Thus, the resulting training data can be fed into n-gram-based classification to evaluate the performance of the model.

2.2 Semi-supervised Self-Labelled Techniques

Many data mining applications have abundant unlabeled examples, whereas labeled ones are costly to obtain. Therefore, semi-supervised learning algorithms have elicited considerable attention. A successful methodology to tackle such problems is self-labeled techniques, which take advantage of a supervised classifier to label instances with unknown class (Triguero et al., 2015). Many machine-learning researchers found that self-labeled techniques can produce considerable improvement in learning accuracy (Li et al., 2013; Ahmed et al., 2015). Self-labeled techniques include two well-known methodologies, namely, co-training and self-training.

The standard co-training (Blum et al., 1998) considers the feature space to be two different conditionally independent views. Each view can train one classifier and then teach the other view to predict the classes (Du et al., 2011; Jiang et al., 2013). By contrast, advanced approaches for co-training are multi-view learning, which does not require explicit feature splits or the iterative mutual-teaching procedure (Zhou et al., 2010; Jin et al., 2014; and Sun et al., 2013).

To the best of our knowledge, self-training was first proposed by Yarowsky et al. (1995) for word sense disambiguation. Self-training attempts to iteratively enlarge the labeled training set by utilizing many unlabeled samples (Li et al., 2005). At the beginning, a classifier is trained with original labeled data. Some highest reliable unlabeled data will then be selected and added incrementally into the labeled training set along with their predicted labels. The procedure is repeated until convergence (Wu et al., 2018). The advantage of self-training is that it does not require a specific assumption, and as a result, it can be used in almost any situation. The self-training approach has been widely used in many domains, including object detection (Rosenberg et al., 2005), face recognition (Roli et al., 2006), and so on.

2.3 Distributed Representation Learning

Recently, neural network models have been used to learn semantic representations for natural language processing tasks, and highly competitive results have been achieved. The idea of distributed representations, also noted as word embeddings, is to project words into low-dimensional continuous-valued vectors. In this manner, variable-sized texts can be encoded into vectors with the same size, and comparison becomes convenient (Mikolov et al. 2013; Li et al. 2016). Among the state-of-the-art neural networks, convolutional neural networks (CNN) models, which were originally invented for computer vision, have also been confirmed effective in using the 1D structure (word order) of text data and have been confirmed effective in capturing the semantics of n-grams of various granularities and achieving state-of-the-art results for many tasks, such as text classification (Kim et al. 2014; Johnson et al. 2015; and Wang et al., 2018), sentence modeling (Kalchbrenner et al. 2014), text-independent speaker verification (Zhang et al., 2018), and so on. Sequential model such as recurrent neural network (RNN) or long short-term memory (LSTM) have also been used for recurrent semantic composition and text classification (Rao et al., 2018; Xia et al., 2018).

Several related work regard fake review detection as a text classification problem. The results of such approaches are encouraging because highly high accuracy has been achieved using only linguistic features (Li et al., 2014). Ren et al. (2016) use a neural network model to study the document-level representation for detecting deceptive opinion spam. The experimental results show that their method outperforms state-of-the-art methods in identifying deceptive opinion spam. Neural models can capture complex semantic information which is difficult to express using traditional discrete manual features. Moreover, neural networks take distributed word embeddings as inputs, which can be trained from a large-scale raw text, thereby alleviating the sparsity of annotated data to some extent. Although neural networks have been proven highly effective in text classification, many data may be needed to be fed into neural models to learn the distributed representation for classification. In this paper, we propose a semi-supervised ensemble learning-based annotation extension scheme and conduct large-scale analysis by training state-of-the-art neural network models on the extended dataset. Thus, our approach is unlike previous studies (Ott et al., 2011; Mukherjee et al., 2012; Li et al., 2014) which rely on small-size data or solicited ground truth fake reviews.

3. Overview of the Approach

To tackle the problem of having few labelled samples and a multitude of unlabeled data in practical application of opinion spam detection, we propose a semi-supervised self-training based scenario in this paper. We use a semi-supervised ensemble-based self-training framework to render an extension to the labelled set and then further investigate the application of neural networks which have been proven highly effective for text classification tasks to learn the continuous representations for training a better classifier. The general view of the proposed framework is plotted in Figure 1.

The formal definition of the problem can be described as follows: Suppose x_i denotes the ith data sample, and ω denotes one class (spam or genuine). L denotes the labelled set with ω known, and U denotes unlabelled set with ω unknown. The number of initial unlabelled data is commonly much larger than that of initial labelled data. The aim is to learn a better classifier C by using L ∪ Ur instead of L alone, where Ur means the extended set of reliable samples from U.

3.1 Annotation Extension

We provide the annotation extension strategy in Algorithm 1. The basic idea of our self-training based annotation extension is as follows. We first use available labelled data to train multiple initial classifier estimators. Each estimator is trained using the features which have been proven effective in previous work. We then use these classifier estimators to find a class label for each unlabelled data item, and some more reliable ones will be extended into Ur. The above strategy is one ensemble based method, and the objective is to find reliable self-learning labelling results from the unlabelled set U to prevent possible performance degradation due to incorrect labeling. The unlabelled data with the highest confidence values are then combined with the originally labelled data to create a newly labelled dataset, namely, $L' = \mathrm{L} \cup U_r$. This extended labelled set L' is further used to train and update the classifiers for labelling unlabelled data in the next iteration. This iterative process continues until a stopping condition is met. To train one sufficiently good initial classifier, the stopping criterion is satisfied when no reliable sample (the reliability score is bigger than τ_r) is selected into U_r.

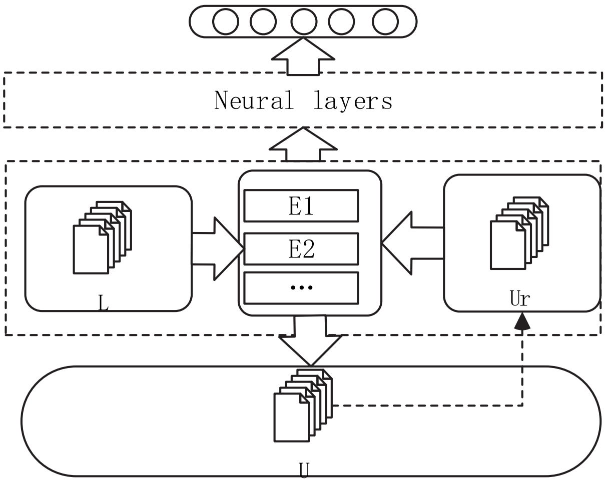


Figure 1. Proposed framework based on semi-supervised annotation extension and neural networks (we use multiple estimators, denoted as E1, E2, and so on, to compute the reliability of samples in the unlabeled set.).

Algorithm 1. Annotation extension algorithm

```
Initialize: L = labelled review set, U = unlabelled review set, $U_r = \Phi$
While stopping criteria not met do
    On training set L ∪ $U_r$, train the initial classifier CI.
    For each unlabelled review u in U:
        Use CI to predict its label ω(u);
        Compute the reliability $R(u)$;
        If $R(u) \geq \tau_r$, then
            Add u to $U_r$, namely, $U_r = U_r \cup u$,
            Delete u from U, namely, $U \leftarrow U - u$;
End while
```

To build a good initial classifier, and most importantly, to find reliable data from the unlabeled set U, one of the choices is the ensemble method because many estimators, which are available for such kinds of methods, can provide a way to compute the reliability score. The goal of the ensemble method is to combine predictions of several base estimators built with a given learning algorithm to achieve better generalizability/robustness compared with that of using a single estimator. On the average, the combined estimator is usually better than any of the single base estimators because its variance is reduced.

We mainly use extra trees (ET) (Breiman et al., 2001; Geurts et al., 2006) as the initial classifier in this paper. ET classifier is the advance of random forest (RF) classifier in the averaging ensemble methods. Similar to RF, a random subset of candidate features is used. Instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature, and the best of these randomly-generated thresholds is selected as the splitting rule in extremely randomized trees. In Section 5, we render an experimental comparison between ET and RF in the selection of initial classifiers, and the results show that the former achieves a better performance.

Moreover, the ratio of two classes of the extended samples after each iteration may be skewed. Imbalanced data often produce poor prediction models (Chawla et al., 2004). Motivated by the study of Drummond and Holte (2003), we use an under-sampling strategy and select a subset of instances, which have the biggest reliability scores, the subset size is equal to that of the minority class from the majority class, and we combine it with the minority class to generate a balanced class distribution data (Note 2).

3.2 Reliability Score

Applying unlabeled data in semi-supervised self-training is beneficial, but in some cases, it may degrade the classifier's performance if they are incorrectly labeled as an improper class by the initial classifier (Piroonsup et al., 2018; Levatić et al., 2017). Some studies try to avoid this issue mainly by post-processing, e.g., self-training with editing (Zhou et al., 2005), or applying a noise-filtering method to remove the mislabeled data (Triguero et al., 2014). However, (Piroonsup et al., 2018) argued that these methods still have some side effects, which result in the feeding of incorrect data into the training process for the final classifier.

To tackle this problem, we use the multiple estimators to compute the reliability score for each sample, since we use ET mechanism to extend the labelled from unlabelled set. Thus the prediction of one sample will be more reliable if the predictions of these classifiers are consistent. Standard deviation can then be used to measure the reliability. Thus, the reliability score can be defined as follows:

$$R_u = 1 - \sqrt{\frac{1}{k-1}\sum_{j=1}^{k}(E_j(u) - M(u))^2} \tag{1}$$

where $E_j(u)$ is the prediction for sample u returned by the jth estimator, and $M(u)$ is the prediction for u returned by the ensemble (i.e., the average of the predictions across all trees). This variance measure has been previously used in the context of bagging, where it performed the best among various approaches for estimating the reliability of regression predictions (Bosni et al., 2008; Levatić et al., 2017).

Once the reliability scores for all samples from the unlabelled set are calculated, they will be sorted by the reliability score in descending order, and the reliable ones are chosen. In Section 5, we further demonstrate the relation of performance with respect to the reliability threshold τ_r.

3.3 Feature Encoding

To train the initial classifier, we concatenate three kinds of features, which have been proven effective in opinion spam detection tasks, for the ensemble based classifier to generate labelled samples.

(1) F1: N-gram

Previous work has shown that bag-of-words is effective in detecting domain-specific deception (Ott et al., 2011; Mihalcea et al., 2009). We use B+, i.e., the union of unigram and bigram as feature F1. Such approach has exhibited better performance than having unigram and bigram alone in related work (Ott et al., 2011).

(2) F2: Linguistic Inquiry and Word Count (LIWC)

LIWC (Tausczik et al., 2010) is a popular computerized automated text analysis tool. It has been used to detect thoughts, feelings, personality, and motivations from the words used in everyday language. This tool has also been adopted to analyze deception (Hancock et al., 2007; Mihalcea et al., 2009). A combined classifier with both n-gram and psychological deception features achieves nearly 90% accuracy on this task (Ott et al., 2011). Although other features have been considered in past deception detection work, early experiments found that LIWC features performed the best (Zhou et al., 2004). Based on Ott et al. (2011), we use features derived from the LIWC output and construct one feature dimension for each LIWC dimension.

(3) F3: Deep Syntax

Previous studies presented interesting correlations between the frequency distribution of syntax features, such as POS tags in a text and the genre of the text (Rayson et al., 2002), which may not be clear to human judges. Ott et al. (2011) found that although POS tags are effective in detecting fake product reviews, they are not as effective as words. Instead, Feng et al., (2012) argued that syntactic stylometry adds a certain unconventional angle to prior literature for deception detection. They used four different datasets, including that for product review spam, to demonstrate that features driven from context-free grammar parse trees consistently improve detection performance over several baselines based only on shallow lexicon-syntactic features. Thus, we use deep syntax feature as feature F3 in the annotation extension algorithm. We apply the Stanford parser (Chen et al., 2014) (Note 3) to parse sentences. In the work of Feng (2012), they experimented on four different encodings of production rules, and the results show that the lexicalized production rules, ˆr∗, exhibit better performance; thus, we use ˆr∗ in this study, which indicates lexicalized production rules combined with the grandparent node, e.g., PRPˆNP 4 → “you.”

4. Neural Models

We train several state-of-the-art neural models to examine the performance of semi-supervised annotation extension as neural networks have achieved promising performance in text categorization tasks. Moreover, for supervised learning tasks, neural methods obviate feature engineering by translating the data into compact intermediate representations similar to principal components and derive layered structures that remove redundancy in representation. In the following, we describe the neural models for training and comparison.

4.1 BiLSTM-RNN

We use one variant of recurrent neural networks, namely, Bi-LSTM (Bidirectional Long Short-Term Memory) (Graves et al., 2013) for review spam classification. Bi-LSTM allows us to look ahead by employing a forward LSTM, which processes the sequence in chronological order, and a backward LSTM, which processes the sequence in reverse order. The output at a given time step is the concatenation of the corresponding states of the forward and backward LSTM as shown in Figure 2.

To build the classifier, we use Softmax, which is calculated as follows:

$$\mathrm{y}(a) = \mathrm{softmax}(W_s a + b_s) \tag{2}$$

$$softmax_i = \frac{\exp(x_i)}{\sum_{j=1}^{C} \exp(x_j)} \tag{3}$$

where $W_s \in R^{2*d}$ is the classification matrix and transforms composition vector a to a real-valued vector, whose length is class number C (2 as there are two classes). Softmax converts real values to conditional probabilities. We use cross-entropy error between the gold sentiment distribution $W_s \in t^{\mathrm{i}} \in R^{C\times 1}$ and predicted sentiment distribution $y^{\mathrm{i}} \in R^{C\times 1}$ as the loss function, which is expressed as follows:

$$\mathrm{loss}(\theta) = \sum_{i\in T} \sum_{j=1}^{C} t_j^i \cdot \log\left(y_j^i\right) + \lambda||\theta||^2 \tag{4}$$

where T is the training data. The loss is a function of parameter θ. All word vectors are stacked in the word embedding matrix $L \in R^{d\times|V|}$, where $|V|$ is the size of the vocabulary. Back propagation is employed to propagate the errors from the top to the leaf nodes. Derivatives of parameters are used to update the parameters. We train our model to minimize cross-entropy loss using stochastic gradient descent and set the learning rate at 0.03. We use the Adam update rule, mini-batches of size 10, and early stopping with a patience of 10.

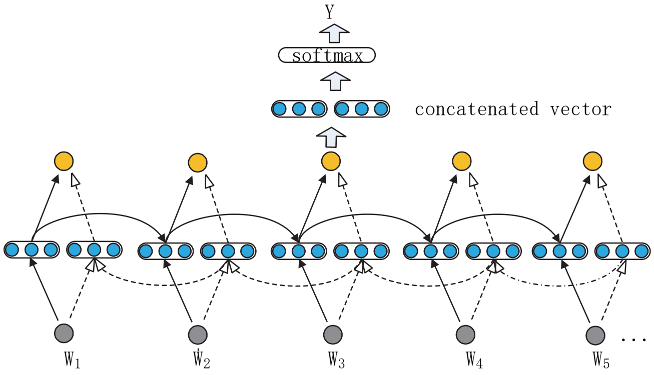


Figure 2. BiLSTM-RNN

4.2 TextCNN

CNNs were originally invented for computer vision (Lecun et al. 2015). These models have been subsequently confirmed effective in classifying 1D structure (word order) text data and have achieved promising results for many tasks (Kalchbrenner et al., 2014). Kim et al. (2014) found that a simple CNN with slight hyper-parameter tuning and static vectors achieved excellent results in sentence level classification tasks. Similar to the RGB-channel in image processing, each filter in text classification is composed of a local patch of lower-level features into higher-level representation. In this paper, we also leverage different convolution window sizes given that CNN has been confirmed effective in capturing the semantics of n-grams of various granularities in text.

We plot the model structure used in this paper in Figure 3. Five layers are observed from the bottom to the top: lookup layer, convolution layer, pooling layer, nonlinear layer, and average layer. As shown in Figure 3, we try CNN with multiple convolutional filters of different widths of 1, 2, and 3 (Tang et al., 2015) to produce sentence representation. Each convolution filter is conducted on the sentence vector by sliding convolution with the step length being 1. Each convolution filter consists of a list of linear layers with shared parameters. Formally, suppose that the input for the linear layer is $I_c = [e_i; e_{i+1}; \dots, e_{i+l_{ic-1}}] \in R^{d \cdot l_{ic}}$, which is the concatenation of word embeddings in a fixed-length input window size l_{ic}. $W_c \in R^{l_{oc} \times d \cdot l_{ic}}$ is the filter, and $b_c \in R^{l_{oc}}$ is the bias, where l_{oc} is the output length of nonlinear layer. The output of a non-linear layer L4 is then generated as follows:

$$O_{C=} f(p(W_C \cdot I_C + b_c)) \quad (5)$$

where p is average pooling function, and f is the non-linear function. We use hyperbolic tangent in this study, but it can naturally incorporate other functions, such as ReLU and the variants.

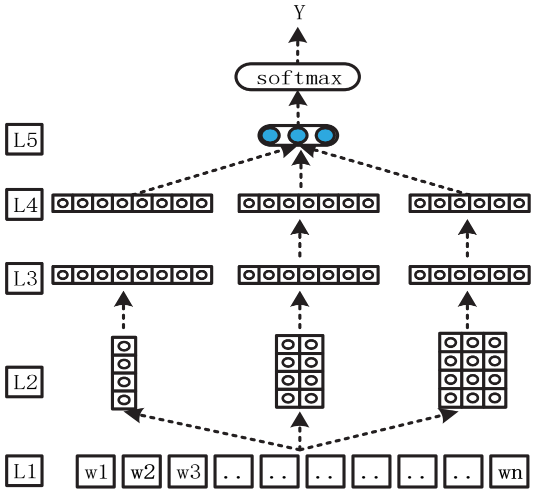


Figure 3. TextCNN. L1: Lookup layer; L2: Convolution layer; L3: Pooling layer; L4: Nonlinear layer (tanh in this paper); and L5: Average layer

4.3 HierAtteNet (Hierarchical Attention Network)

Yang et al. (2017) argued that for document classification, not all words contribute equally to the semantic representation of the sentence, and not all sentences contribute equally to the semantic representation of the entire document. We then investigate how Hierarchical Attention Network (HierAtteNet for short) work in our extended spam review dataset. We use GRU+attention mechanism to code the sentence vector and the word vector. It has six layers: 1) Embedding. For each word in the review, we use their word embedding form by using pre-trained lookup table. 2) Word Encoder. Word-level bidirectional GRU (Bahdanau et al., 2014) is used to code the word sequence in review text and determine hidden representation of words. 3) Word Attention. Word-level attention obtains important information in a sentence. 4) Sentence Encoder. Sentence-level bi-directional GRU obtains the hidden representation of sentences. 5) Sentence Attention. Sentence-level attention selects the important sentence among sentences. 6) Full Connection and Softmax. In the end, the learned representation of review document vector can be used for spamicity classification. Figure 4 depicts layers 4–6, and we provide the corresponding computation process. We omit that of layers 1–3 as the processes are highly similar with layers 4–6.

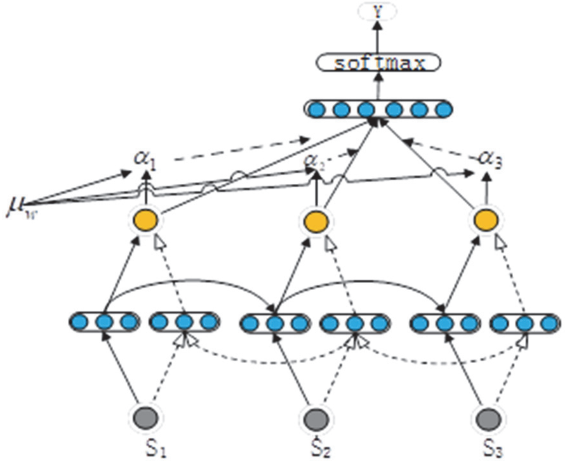


Figure 4. HierAtteNet

In this mechanism, the review vector can be computed as follows:

$$\mathrm{h}_{it} = [\overleftarrow{GRU(S_{it})}, \overrightarrow{GRU(S_{it})}] \tag{6}$$

$$u_{it} = \tanh(W_w h_{it} + b_w) \tag{7}$$

$$\alpha_{it} = \frac{\exp(u_{it}^T \mu_w)}{\sum_{t=1}^{T} \exp(u_{it}^T \mu_w)} \tag{8}$$

$$\mathrm{v} = \sum_{t=1}^{T} \alpha_{it} h_{it} \tag{9}$$

$$\mathrm{p} = \mathrm{softmax}(W_c \mathrm{v} + b_c) \tag{10}$$

In the above equations, the input S_{it} denotes the embedding form of sentence. After calculating h_{it}, the normalized importance weight αit for each$h_{it}$is obtained through a Softmax function. The review vector v is computed as a weighted sum of the word annotations h_{it} based on the weights αit. In the end, the learned representation of review document vector v can be used for review spam classification. The parameter context vector μw is randomly initialized and jointly learned during the training process. We use cross entropy as training loss and set max sentence length to 100 tokens, decay rate to 0.9, and optimizer to "rmsprop" in this paper.

5. Experimental Results and Analysis

5.1 Experiment Setup

Dataset and settings. As discussed previously, Ott et al. (2013) introduced an opinion spam corpus, which is a highly reliable ground-truth dataset as they hire turkers to write fake reviews; thus, the dataset can simulate the real situation to the maximum extent. In this paper, we call this corpus the Ott dataset and use it as the labeled data in our work. In the Ott dataset, the reviews, which are approximately hotels, is composed of 400 genuine and 400 fake reviews, totaling 800 reviews for each polarity (negative or positive). To remain consistent with their work, we also use hotel reviews as the unlabeled data in this study, whereas we collected up to 20,000 unlabeled hotel reviews from TripAdvisor for each polarity; thus, the number is 25 times that of the original Ott dataset. In Table 1, we provide the description of the original Ott dataset and that of the unlabeled dataset which is used for annotation extension. Detecting review spam is a challenging task because no one exactly knows the amount of spam in existence (Lim, et al., 2010), so the number of genuine reviews and the number of fake reviews in the unlabeled dataset is unknown and filled with "--" in Table I.

Table 1. Dataset description

Name	Polarity	#Genuine	#Fake	#Total
Ott	Negative	400	400	800
Ott	Positive	400	400	800
Unlabelled	Negative	--	--	20,000
Unlabelled	Positive	--	--	20,000

All the tests are performed on the Ott dataset. We use 80% of the Ott data and all the unlabelled data for self-training, keep remaining 20% of the Ott dataset as the test set, shuffle all the data in this dataset randomly, and use five-fold cross-validation to examine the performance.

Baselines and Evaluation Metrics. We compare against the following methods as our baselines, which consist of some existing work and neural network-based models: (1) Ott et al.'s method, which uses SVM classifier and B++LIWC features (Note 4); (2). Ren's work (Ren et al., 2016), which is also a neural network-based approach for opinion spam detection; (3). Self-trained classifier. We use scikit-learn implementation (Note 5)of ET as classifiers and use three kinds of features shown in Section 3; and (4). Neural models described in Section 4. We trained these models on the finally extended dataset to examine if annotation extension results in improvements. We use pre-trained static vectors trained by Google word2vec toolkit (Note 6) from all the datasets (including both the labelled and unlabelled set) as the word embedding representation (embedding size is 300), and we adopt a non-static mode in the training process to update the parameter dynamically.

We report the results of standard measures, namely, accuracy, precision, recall, and F1 measure on the test set. Suppose TP denotes true positives (Note 7), FN denotes false negatives, FP denotes false positives, and TN denotes true negatives. These measures can be calculated as follows:

$$Accuracy = \frac{TP+TN}{P+N} \tag{11}$$

$$Precision = \frac{TP}{TP+FP} \tag{12}$$

$$Recall = \frac{TP}{P} \tag{13}$$

$$F_1 - score = \frac{2*Precision*Recall}{(Precision+Recall)} \tag{14}$$

5.2 Results and Analysis

We provide the classification results of several comparative models and existing baselines for a general performance comparison. Table 2 (a) and (b) illustrate the results in two different kinds of datasets, namely, negative and positive, respectively. The results of several significance tests for paired t-test are also given in Table 3. We use ET as the ensemble classifiers, use 80% of the original Ott dataset as the initial training set, set reliability threshold to 0.625, and set the number of estimators to 200 because these settings are optimal as shown in the following experiments.

Our observations from Tables 2 and 3 are as follows:

First, by using our annotation extension strategy, nearly all neural network models gain performance improvements on the extended dataset. The corresponding significance test shown in Table 3 implies that p-value of "AE+NN>NN" is less than 10−3. This finding suggests that using a typical small amount of labelled spam data which are readily available but expensive to obtain with a large amount of unlabelled data for training is feasible in opinion spam detection. Furthermore, it also suggests that semi-supervised self-training strategy is not only helpful in learning one well-learned classifier but also beneficial in forming a semi-labelled dataset which can be used further for training of neural network models.

Second, AE+NN>AE+Ott et al. (SVM) and the significance test show the p-value is less than 10^{-2}. This suggests that neural networks are better than SVM for the self-learning extension dataset. We attribute this improvement to the generality capability of neural networks. As shown in Table 2, the performance of “AE+Ott et al. (SVM)” is better than “Ott et al. (SVM),” which implies that the self-learning based extension strategy itself is effective for other kinds of classifiers such as SVM, which has been used in (Ott et al., 2013).

Third, we also render a comparison with some existing work in Table 2. We compare our approach with the work of Ren et al. (Ren et al., 2016), which also uses neural networks to detect opinion spam, and we re-implemented their work. The results show that our approaches are significantly more effective.

We can also find that the spam detection of the negative polarity dataset is generally more difficult than that of the positive polarity dataset. However, the above trends can still be observed for both datasets.

Table 2. Classification performance comparison. “AE” means annotation extension, and we feed the properly extended dataset based on self-learning into the neural network models.

(a) Positive dataset

Methods	Acc.	Pre.	Rec.	F1
Ott et al. (SVM) (Ott. 2013)	89.30	88.90	89.80	89.30
Ren et al. (2016)	90.65	90.28	91.57	90.92
AE+Ott et al. (SVM)	90.42	91.21	90.49	90.85
Extra-trees	89.86	89.73	90.14	89.93
Bi-LSTM	88.95	88.68	89.32	89.00
AE+ Bi-LSTM	90.76	91.63	90.97	91.30
TextCNN	89.12	89.34	88.93	89.13
AE+TextCNN	91.83	92.51	90.95	91.72
HierAtteNet	89.82	89.74	89.98	89.86
AE+HierAtteNet	92.37	92.64	91.28	91.95

(b) Negative dataset

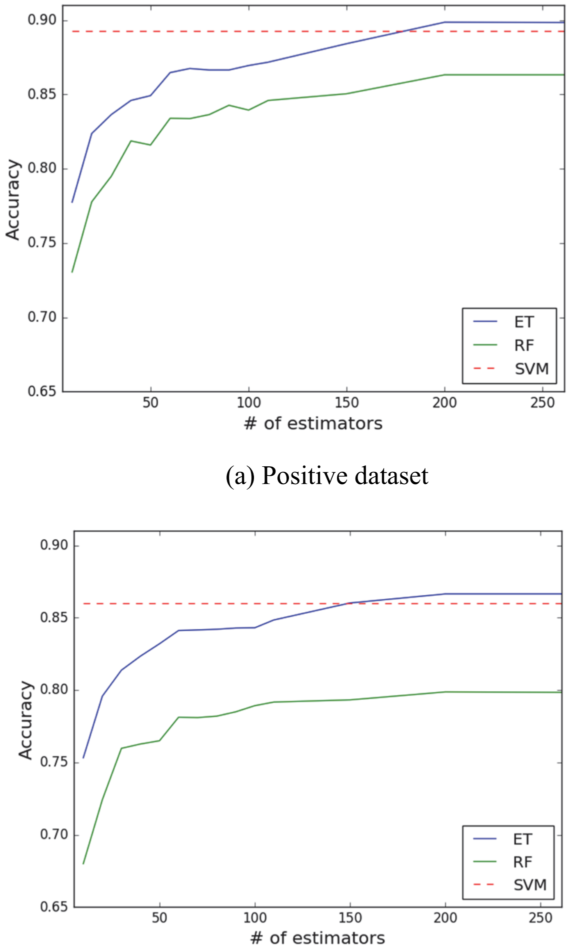
Methods	Acc.	Pre.	Rec.	F1
Ott et al. (SVM) (Ott. 2013)	86	85.6	86.5	86.1
Ren et al. (2016)	86.57	87.2	86.3	86.75
AE+Ott et al. (SVM)	87.2	87.34	87.2	87.27
Extra-trees	86.65	86.93	86.2	86.58
Bi-LSTM	85.43	85.89	85.2	85.56
AE+ Bi-LSTM	88.15	87.42	88.9	88.15
TextCNN	85.54	85.68	85.5	85.60
AE+TextCNN	87.78	88.39	87.2	87.80
HierAtteNet	86.24	86.54	85.8	86.16
AE+HierAtteNet	89.26	89.54	89.3	89.43

Table 3. Paired t-test results

Comparison	Metrics and datasets	p-value
AE+NN>NN	All	$< 10^{-3}$
AE+NN> AE+Ott et al. (SVM)	All	$< 10^{-2}$
AE+NN> Ott et al. (SVM) (2013)	All	$< 10^{-3}$
AE+NN> Ren et al. (2016)	All	$< 10^{-3}$

5.3 Initial Self-learning Classifiers

In this subsection, we conduct experiments to see the effect of the selection of the initial classifiers and the number of estimators on the self-learning performance. We then find the suitable number of estimators based on the experimental results in training an effective initial self-learning classifier. Many previous work showed that inefficient initial classifiers can introduce incorrectly labelled data. If the mislabelled data are used to train the subsequent classifiers, then the performance will be affected adversely.



(a) Positive dataset

(b) Negative dataset

Figure 5. Effect of # of estimators

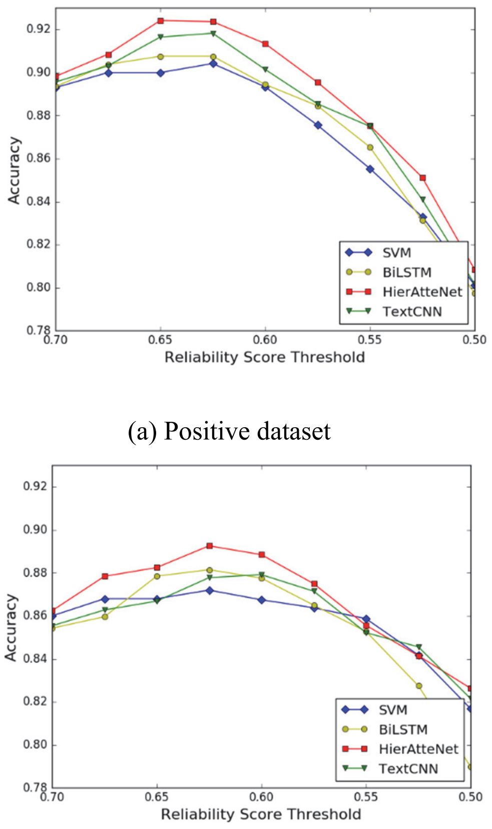
Based on this consideration, we rendered a comparison with several classifiers, including RF, ET, and SVM (with linear kernel). We also varied the number of estimators in the RF and ET classifier to see the performance difference. The results are shown in Figure 5, where x-axis is the number of iterators, and vertical axis is the accuracy (Note 8). In this experiment, we use all the 80% labelled data as the training set and the remaining 20% as the test set.

As shown in Figs. 5 (a) and (b), with the increase of estimator number, the performances of both ET and RF increase correspondingly. ET generally perform better than RF. We can observe that when the number of estimators is around 200, the accuracy for both classifiers begin to stabilize. We also find that ET with enough number of estimators (e.g., 200) are slightly better than SVM. Recall that by using ET classifier, calculating the reliability score based on standard deviation of these estimators is highly convenient, as shown in Section 3.

5.4 Varying the Reliability of Labelled Training Data

In this subsection, we conduct experiments to see the performance of the proposed approach with respect to the reliability score threshold τ_r in Algorithm 1. For same τ_r, we use 80% of the labelled data for self-training and keep the remaining 20% as the test set. Five-fold cross-validation is used to examine the performance. Figs. 6 (a) and (b) illustrate the results in two different kinds of datasets, namely, negative and positive, respectively. The x-axis is the reliability score threshold τ_r, and the y-axis is the accuracy. For our binary classification problem (class label is 1 or 0), the largest possible prediction deviation is 0.5, which means the worst situation. In the experiments, we find that when reliability score threshold is greater than 0.7 (namely, the standard deviation of classification threshold is smaller than 0.3), highly few unlabelled data are selected, and the performance remains nearly the same as training with the original labelled datasets. So in both figures, the range of τ_r is set between 0.7 and 0.5, which also means the corresponding range of classification standard deviation is between 0.3 to 0.5.

These two figures show that at the beginning, namely, when the reliability score threshold is highly high, e.g., approximately 0.7 to 0.625 in the figures, the performance improves with the decrease of the threshold, and the performance reaches the peaks when the threshold is approximately 0.625 to 0.6. Afterward, with the subsequent decrease of reliability score threshold, the performance begins to degrade. We argue that the underlying reason for above trends is that high reliability score threshold means strict constraints on the classification consistence of estimators. Thus, few unreliable data may be eligible to be selected into the extended training set, and the performance may remain unchanged or slightly improve compared with training using only the original labelled data. More reliable unlabelled data are selected with the moderate decrease of the reliability threshold. Thus, we can see a performance gain from the curves. By contrast, when the reliability threshold is highly low, e.g., lower than 0.55 in experiments, the classifiers may be affected by increasingly more unreliable data, which will worsen the classification performance. In related work (Bosni et al., 2008; Levatić et al., 2017), they also argued that the disadvantage of self-learning is that if samples of wrong classification are added to the training set, the errors that they render in the subsequent training process will be increasingly deeper, and the other samples tend to render mistakes. Thus, the performance of self-training can deteriorate. Their findings may also help explain our experimental results.



(a) Positive dataset

(b) Negative dataset

Figure 6. Effect of reliability threshold τ_r

6. Conclusions and Future Work

The expensiveness of spam review corpus construction process may render a fully labelled training set infeasible, whereas unlabelled data is relatively inexpensive to acquire. Thus, in this work, we propose an opinion spam detection method based on annotation extension and neural models. Our work has demonstrated that the proposed semi-supervised based annotation extension strategy has considerable practical value. Improvements can be achieved by training state-of-the-art neural networks, particularly due to their generalization capability, and translating the data into intermediate compact representations without manual feature engineering.

Our work suggests that semi-supervised self-training is an efficient method for learning with few labelled data and a large set of unlabelled data. In addition, our strategy is feasible for the opinion spam detection task. For the future work, as our preliminary work has shown that the reliability of unlabelled data plays an important role in the performance of such kind of methods, the possible directions may focus on finding reliable labelled data or developing more powerful initial classifiers in the self-learning process.

Acknowledgments

The research was supported by the National Natural Science Foundation of China (61672192 and 61572151).

References

Ahmed, I., Ali, R., Guan, D., Lee, Y. K., Lee, S., & Chung, T. C. (2015). Semi-supervised learning using frequent itemset and ensemble learning for sms classification. *Expert Systems with Applications*, *42*(3), 1065-1073. http://dx.doi.org/10.1016/j.eswa.2014.08.054

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. Computer Science, *arXiv preprint arXiv:1409.0473.*

Blum, A., & Mitchell, T. (1998). Combining labelled and unlabelled data with co-training. *in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory,* ACM, New York, NY, USA, 1998, pp. 92-100.

Bosnić, Z., & Kononenko I. (2008). Comparison of approaches for estimating reliability of individual regression predictions. *Data & Knowledge Engineering*, *67*(3), 504-516. http://dx.doi.org/10.1016/j.datak.2008.08.001

Breiman L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.

Cardoso E. F. et al. (2018). Towards automatic filtering of fake reviews, *Neurocomputing*, pp. 106-116. http://dx.doi.org/10.1016/j.neucom.2018.04.074

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, *6*(1), 1-6. http://dx.doi.org/10.1145/1007730.1007733

Chen, D., & Manning, C. (2014). A Fast and Accurate Dependency Parser using Neural Networks. *Conference on Empirical Methods in Natural Language Processing*, pp. 740-750.

Chirita, P. A., Diederich, J., & Nejdl, W. (2005). MailRank: Using Ranking for Spam Detection, *Conference on Information and Knowledge Management*, pp. 373-380.

Drummond, C., & Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: why under-sampling beats oversampling. *In Proceedings of the ICML Workshop on Learning from Imbalanced Datasets II*, pp. 1-8.

Du J. et al. (2011). When does co-training work in real data?. *IEEE Trans. Knowl. Data Eng*, volume 23, issue 5, pp. 788-799.

Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R. (2013). Exploiting Burstiness in Reviews for Review Spammer Detection, *ICWSM 2013*, pp. 175-184.

Feng, S., Banerjee, R., & Choi, Y. (2012). Syntactic stylometry for deception detection, *Meeting of the Association for Computational Linguistics: Short Papers*, pp. 171-175.

Geurts, P. et al. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3-42.

Graves, A, Mohamed, A., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. I*EEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645-6649.

Hai, Z., Zhao, P., Cheng, P., Yang, P., Li, X. L., & Li, G. (2016). Deceptive Review Spam Detection via Exploiting Task Relatedness and Unlabelled Data, *Conference on Empirical Methods in Natural Language Processing*, pp. 1817-1826.

Heydari, A., Tavakoli, M. A., Salim, N., & Heydari, Z. (2015). Detection of review spam: a survey. *Expert Systems with Applications*, *42*(7), 3634-3642. http://dx.doi.org/10.1016/j.eswa.2014.12.029

Jeffrey, T., Hancock, L. E., Curry, S. G., & Michael, W. (2007). On lying and being lied to: a linguistic analysis of deception. *in computer-mediated communication. Discourse Processes*, *45*(1), 1-23.

Jiang Z. et al. (2013). A hybrid generative/discriminative method for semi-supervised classification, *Knowledge Based System*, *37,* 137-145. http://dx.doi.org/10.1016/j.knosys.2012.07.020

Jin G., & Raich R. (2014). Hinge loss bound approach for surrogate supervision multi-view learning, *Pattern Recognition. Letters*, *37,* 143-150. http://dx.doi.org/10.1016/j.patrec.2013.06.008

Jindal, N., & Liu, B. (2008). Opinion spam and analysis, *International Conference on Web Search and Data Mining*, pp. 219-230.

Johnson, R., & Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks, *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pp. 103-112.

Kalchbrenner N. et al. (2014). Sequence to sequence learning with neural networks, *In Advances in Neural Information Processing Systems,* pp. 3104-3112

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1746-1751.

Kolari, P., Java, A., Finin, T., Oates, T., & Joshi, A. (2006). Detecting Spam Blogs: A Machine Learning Approach. American Association for Advancement of Artificial Intelligence, 2006, pp. 1351-1356.

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep Learning. *Nature, 521*, 436-444.

Levatić, J., Ceci, M., Kocev, D., & Džeroski, S. (2017). Self-training for multi-target regression with tree ensembles. *Knowledge-Based Systems*, *123*(C), 41-60. http://dx.doi.org/10.1016/j.knosys.2017.02.014

Li Huayi, et al. (2014). Spotting Fake Reviews via Collective Positive-Unlabelled Learning. *In Proceedings of the 2014 IEEE International Conference on Data Mining (ICDM '14)*. IEEE Computer Society, Washington, DC, USA, pp. 899-904.

Li, J., Ott, M., & Cardie, C. (2013). Identifying Manipulated Offerings on Review Portals. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1933-1942.

Li Jianqiang, & Li Jing, et al. (2016). Learning distributed word representation with multi-contextual mixed embedding. *Knowledge-Based Systems, 106,* 220-230. http://dx.doi.org/10.1016/j.knosys.2016.05.045

Li, M., et al. (2005). Setred: Self-training with Editing, *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, 3518, Springer Berlin Heidelberg, pp. 611-621.

Li, H., Chen, Z., Mukherjee, A., Liu, B., & Shao, J. (2015). Analyzing and Detecting Opinion Spam on a Large-scale Dataset via Temporal and Spatial Patterns, *Proceedings of Ninth International AAAI Conference on Web and Social Media*, pp. 634-637.

Li, J., Ott, M., Cardie, C., & Hovy, E. (2014). Towards a General Rule for Identifying Deceptive Opinion Spam. *Meeting of the Association for Computational Linguistics*, pp. 1566-1576.

Li, Luyang et al. (2017). Document representation and feature combination for deceptive spam review detection. *Neurocomputing*, *254,* 33-41. http://dx.doi.org/10.1016/j.neucom.2016.10.080

Lim, E. P., Nguyen, V. A., Jindal, N., Liu, B., & Lauw, H. W. (2010). Detecting product review spammers using rating behaviors. *ACM International Conference on Information and Knowledge Managemen*t, pp. 939-948.

Luca, M., & Zervas, G. (2013). Fake it till you make it: Reputation, competition, and Yelp review fraud. Harvard Business School NOM Unit Working Paper.

Michael, L. (2010). Reviews, reputation, and revenue: the case of yelp.com. *Harvard Business School Working Papers*, USA, November.

Mihalcea, R., & Strapparava, C. (2009). The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language. *Proceedings of the, Meeting of the Association for Computational Linguistics and the, International Joint Conference on Natural Language Processing of the AFNLP*, Short Papers, pp. 309-312.

Mikolov, T, Sutskever, I., & Chen, K, et al. (2013). Distributed representations of words and phrases and their compositionality. *International Conference on Neural Information Processing Systems*, pp. 3111-3119.

Mukherjee, A. (2015). Detecting Deceptive Opinion Spam using Linguistics, Behavioral and Statistical Modeling. *Association for Computational Linguistics Tutorials,* Beijing, China, Jul. pp. 21-22.

Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., & Castellanos, M., et al. (2013). Spotting opinion spammers using behavioral footprints. *ACM SIGKDD International Conference on Knowledge Discovery and Data*

Mining, pp. 632-640.

Mukherjee, A., Liu, B., & Glance, N. (2012). Spotting Fake Reviewer Groups in Consumer Reviews. *Proceedings of the 21st international conference on World Wide Web*, pp. 191-200.

Ott, M, Choi, Y., & Cardie, C, et al. (2011). Finding deceptive opinion spam by any stretch of the imagination. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies,* pp. 309-319.

Ott, M., Cardie, C., & Hancock, J.T. (2013). Negative deceptive opinion spam. *Proceedings of NAACL-HLT*, pp. 497-501.

Piroonsup, N., & Sinthupinyo, S (2018). Analysis of training data using clustering to improve semi-supervised self-training. *Knowledge-Based Systems*, *143*, 65-80. http://dx.doi.org/10.1016/j.knosys.2017.12.006

Rao G. et al. (2018). LSTM with sentence representations for document-level sentiment classification. *Neurocomputing*, *308,* 49-57. http://dx.doi.org/10.1016/j.neucom.2018.04.045

Rayana, S., & Akoglu, L. (2015). Collective Opinion Spam Detection: Bridging Review Networks and Metadata. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 985-994.

Rayson, P., Wilson, A., & Leech, G. (2002). Grammatical word class variation within the british national corpus sampler. *Language & Computers*, pp. 295-306.

Ren, Y., & Zhang, Y. (2016). Deceptive Opinion Spam Detection Using Neural Network. *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 140-150.

Ren, Y, & Ji, D. (2017). Neural Networks for Deceptive Opinion Spam Detection: An Empirical Study., *Information Sciences*, 385-386, pp. 213-224. http://dx.doi.org/10.1016/j.ins.2017.01.015

Roli, F., & Marcialis, G. L. (2006). Semi-supervised PCA-Based Face Recognition Using Self-training. *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Worksh*ops, SSPR 2006 and SPR 2006, Hong Kong, China, August 17-19, pp. 560-568.

Rosenberg C., et al., (2005). Semi-supervised self-training of object detection models. *in: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*, *1*, 29-36.

Santosh, K. C., & Mukherjee, A. (2016). On the Temporal Dynamics of Opinion Spamming: Case Studies on Yelp, *International Conference on World Wide Web*, pp. 369-379.

Savage, D., Zhang, X., & Chou, P, et al. (2015). Detection of opinion spam based on anomalous rating deviation. *Expert Systems with Applications, 42*(22), 8650-8657. http://dx.doi.org/10.1016/j.eswa.2015.07.019

Shen, G., Gao, B., Liu, T.-Y., Feng, G., Song, S., & Li, H. (2006). Detecting link spam using temporal information. *Sixth International Conference on Data Mining*, pp. 1049-1053.

Sun, S. (2013). A survey of multi-view machine learning. *Neural Computing & Applications*, *23*, 2031-2038. http://dx.doi.org/10.1007/s00521-013-1362-6

Tang, D., Qin, B., & Liu, T. (2015). Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. *Conference on Empirical Methods in Natural Language Processing*, pp. 1422-1432.

Tausczik, Y. R., & Pennebaker, J. W. (2010). "The psychological meaning of words: liwc and computerized text analysis methods". *Journal of Language & Social Psychology, 2*(1), 24-54.

Triguero, I. et al. (2015). Self-labelled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge & Information Systems, 42*(2), 245-284.

Triguero, I., et al. (2014). On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. *Neurocomputing*, *132,* 30-41. http://dx.doi.org/10.1016/j.neucom.2013.05.055

Wang, Peng et al. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short textclassification. *Neurocomputing*, *174*, Part B, pp. 806-814.

Wang, X., Liu, K., & He, S. et al. (2016). Learning to Represent Review with Tensor Decomposition for Spam Detection. *EMNLP*, pp. 866-875.

Wang, G., Xie, S., Liu, B., & Yu, P. S. (2011). Review Graph Based Online Store Review Spammer Detection. *IEEE International Conference on Data Mining*, pp. 1242-1247.

Wu, D., Shang, M. S., Luo, X., Xu, J., Yan, H. Y., & Deng, W. H., et al. (2018). Self-training semi-supervised classification based on density peaks of data. *Neurocomputing*, *275*, 180-191.

http://dx.doi.org/10.1016/j.neucom.2017.05.072

Xia, Wei, et al. (2018). Novel architecture for long short-term memory used in question classification. *Neurocomputing*, *299*, 20-31. http://dx.doi.org/10.1016/j.neucom.2018.03.020

Xu, C., Zhang, J., & Chang K, et al., (2013). Uncovering collusive spammers in Chinese review websites. *ACM International Conference on Conference on Information & Knowledge Management*, pp. 979-988.

Xu, Y., Shi, B., Tian, W., & Lam, W. (2015). A unified model for unsupervised opinion spamming detection incorporating text generality. *International Conference on Artificial Intelligence*, 725-731.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2017). Hierarchical Attention Networks for Document Classification. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480-1489.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *in: Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189-196.

Zhang, Chunlei et al. (2018). Text-Independent Speaker Verification Based on Triplet Convolutional Neural Network Embeddings. I*EEE/ACM Transactions on Audio, Speech and Language Processing*, *26*(9), 1633-1644.

Zhou, Lina et al. (2004). A comparison of classification methods for predicting deception in computer-mediated communication. *Journal of Management Information Systems*, *20*(4), 139-165. http://dx.doi.org/10.1080/07421222.2004.11045779

Zhou, Z. H., & Li, M. (2010). Semi-supervised learning by disagreement. *Knowl. Inf. Syst.*, *24*(3), 415-439. http://dx.doi.org/10.1007/s10115-009-0209-z

Notes

Note 1. http://myleott.com/op_spam

Note 2. Ott's dataset (Ott et al., 2013) is also balanced (50% class distribution). For a detailed analysis of detection in the skewed class distribution, refer to Mukherjee et al. (2013)

Note 3. https://nlp.stanford.edu/software/lex-parser.html

Note 4. B+ means "uingram+bigram"

Note 5. http://scikit-learn.org/

Note 6. https://code.google.com/p/word2vec

Note 7. "Positive" and "negative" here are the commonly used terms in performance evaluation. The former means spam review, and the latter means non-spam in this paper, so they are not sentiment polarity.

Note 8. We use accuracy as the evaluation metric in the following experiments because we found that the accuracy results from various models were nearly equivalent to F-measures on these two balanced datasets.

Copyrights