Front. Comput. Sci., 2015, 9(2): 171–184
DOI 10.1007/s11704-014-4085-7

RESEARCH ARTICLE

Structural information aware deep semi-supervised recurrent neural network for sentiment analysis

Wenge RONG[1,2], Baolin PENG[1], Yuanxin OUYANG (✉)[1,2], Chao LI[1,2], Zhang XIONG[1,2]

1 School of Computer Science and Engineering, Beihang University, Beijing 100191, China
2 Research Institute of Beihang University in Shenzhen, Shenzhen 518057, China

Abstract With the development of Internet, people are more likely to post and propagate opinions online. Sentiment analysis is then becoming an important challenge to understand the polarity beneath these comments. Currently a lot of approaches from natural language processing's perspective have been employed to conduct this task. The widely used ones include bag-of-words and semantic oriented analysis methods. In this research, we further investigate the structural information among words, phrases and sentences within the comments to conduct the sentiment analysis. The idea is inspired by the fact that the structural information is playing important role in identifying the overall statement's polarity. As a result a novel sentiment analysis model is proposed based on recurrent neural network, which takes the partial document as input and then the next parts to predict the sentiment label distribution rather than the next word. The proposed method learns words representation simultaneously the sentiment distribution. Experimental studies have been conducted on commonly used datasets and the results have shown its promising potential.

Keywords sentiment analysis, recurrent neural network, deep learning, machine learning

1 Introduction

With the explosive development of social technology, the information from blogs, forum, product reviews and social media collect a huge volume data of public opinions. People becomes used to get important piece of information from other people during decision making. For example, before purchasing a product, more and more people will firstly have a survey from product reviews to get the basic idea of product quality based on the opinions of existing users. Business enterprises widely keep a close eye on public opinions about their brand, product or service and listen carefully to the public criticisms and suggestions. The quickly accumulated large archive of opinions call for efficient and effective mechanism to extract and analysis them. To meet this challenge, sentiment analysis is widely lauded as new momentous to study the opinions, sentiments, attitudes and emotions expressed in text such as twitters, blogs, production reviews and etc. [1–5].

There are a lot of methods proposed to conduct sentimental analysis. Intuitively, many bag-of-words based models can be applied to solve this kind of problem. The bag-of-words models are widely used in information retrieval and have been proven its success due to its simpleness and robust in implementation. Bag-of-words normally relies on the words occurrence pattern in the document as such it is able to void the language morphology and capture simple patterns on character level. Though it has shown its promising applicability, there exist several challenges in employing it into sentimental analysis.

One of the challenges for bag-of-words oriented methods is to grasp linguistic pattern in sentiment analysis [3,6]. Generally a text contains both syntactical and lexical information, while bag-of-words models normally put the structure information aside, e.g., the words sequence information. As

Received March 2, 2014; accepted April 28, 2014

E-mail: oyyx@buaa.edu.cn

a result, in some cases, though two phrases have same bag-of-words representation, their real meaning could be totally opposite [7]. One of the possible reasons is because relying only on individual words represented as indices in a vocabulary will not be able to obtain the rich relational lexicon structure [8].

To overcome this problem, semantic oriented approaches are brought forward trying to add semantic information into sentiment analysis process. This kind of methods rely on annotation techniques to add polarity scores to the words or phrases in a statement [9]. Though integrating extra semantic information is promising, automatic annotation remains a major challenge which will limit the probability to scale up or transit to another languages.

An alternative type of approaches tries to take advantage of simpleness of bag-of-words while does not rely on outer information to maintain extensibility. This kind of methods utilise statement's lexical and structural information to help calculate the polarity [10]. An typical example is the so-called problem of "atomic units" where the particular attitude of a statement is not dependent on single worlds but appraisal groups [11].

In this paper, inspired by underlying importance of a statement's structure in sentiment analysis process, we proposed a novel approach to improve polarity detection performance by analysing the relationship among the segments within a statement. The basic hypothesis is that while the phrases or sentences within a statement are strongly and directly related to the overall polarity [12], their inner sequence order is also an important kind of hints for sentiment orientation detection [13].

To fully utilise the inter-relation information among segments in a statement, the proposed method employed recurrent neural network (RNN) to enhance the precision of sentiment analysis based on the capability of RNN in structured data predication [14–17], which has been widely used in the domain of natural language processing (NLP) [18]. Built upon RNN, a discriminative model is implemented to learn the inner representation of words, which are believed to contain more information than mere bag-of-words. Furthermore, the proposed model will use extra unlabelled data to conduct unsupervised pre-training, which aims to acquiring weight between input layer and hidden layer and also known as word embedding, and utilize labelled data to run fine-tuning, in which the weights will be trained via using traditional back propagation algorithm in a supervised way. Meanwhile, considering the complicated interaction between components within a statement, it is more reasonable to use a trainable model. As such the semi-supervised mechanism is employed in this research to conduct data training, which has been proven successful in performance efficiency and effectiveness [19].

During the data training process, it is worthwhile to point out that sentiment label distribution at previous time step would also contribute to the distribution at next time [20]. Based on this finding, the proposed RNN model has been slightly modified and compared to classic Elman recurrent neural network architecture [21], it consists of inputs to a set of hidden nodes, a fully connected set of recurrent connections between these hidden nodes, and also a fully connected set of recurrent connections between output nodes which implicitly keep sentiment distribution at previous time step. Therefore, in the semi-supervised dual RNN model, the input is word indices in vocabulary and the output is sentiment label distribution. In order to capitalize the recursion feature of RNN, we split the description words in a statement into different parts and then feed to RNN sequentially. As a result, the last output corresponding to the last input is the final distribution of sentiment label.

The remaining of this paper is organized as follows. The background information about RNN and its application in sentiment analysis will be presented in Section 2. The proposed method will be illustrated in Section 3 and Section 4 will elaborate and discuss the experimental study on different standards datasets. Finally Section 5 will give conclusion and point out possible future research directions.

2 Background

2.1 Sentimental analysis

Sentiment analysis, which makes it possible for users to infer from statements whether or not the overall sentiment is favourable, is a key technology of information gathering for decision making [22]. It has been widely lauded as a new momentous mechanism to understand opinions from large number of on-line documents such as twitters, blogs, production reviews and etc [1–5]. Effectively discriminating sentiment distribution has then become a fundamental challenge and has been attached much importance for both research and application purposes [23–25]. A large portion of study is working on this problem and various approaches and solutions have been proposed and widely used to different application scenarios.

The most widely used approach for sentiment analysis is based on words occurrence pattern in the document, which is usually called “bag-of-words”. Quite a number of unsupervised learning based approaches have been proposed to determine the positivity via the dominant polarity of the opinion words in the given piece of text [12]. By the function of positive/negative indicators in text, a general polarity score is calculated. Obviously, the basis of lexicon based approaches is the generation of sentiment lexicon corpus. In some work, the polarity labels of terms are determined by a set of pre-defined seed words with the consideration of their co-occurrence relationship [26]. Some work learned the dictionary from the semi-structured reviews [27], where the explicit rating and aspect indicators can be used to determine the collect sentiment terms.

Compared with lexicon based approach as discussed above, machine learning based sentiment classification is another approach, which treat sentiment classification as a special case of categorization with two class of positive and negative. In these kind of approaches, feature definition is very important for learning based approaches. In the condition of social media, machine learning based approach for sentiment classification is also widely applied [28].

In this paper, we studied the problem using machine learning and particularly employed RNN to enhance the precision of sentiment analysis. The idea is inspired by the fact that the inner relationship between segments or words within a statement has directly impact on the final polarity detection. Considering the capability of RNN in structured data predication [14–17], we employ it to further capture structural information to support sentiment analysis.

2.2 Recurrent neural network

A neural network is an interconnected group of natural or artificial neurons which use a mathematical or computational model for information processing based on a connectionist approach to computation. In most cases an neural network is an adaptive system which changes its structure based on external or internal information which flows through the network [29]. Neural networks have been used in many applications successfully such as speech recognition [30,31], image processing [32,33], dimension reducing [34] and etc.

However, traditional neural network does not get well accepted performance in structure data prediction where its input is variable. To overcome this limit, recently RNN has been proposed and achieved great success [14–17]. All the achievement maybe due to its important feature called recursion, through which the network can be implicitly deeper than traditional neural network [35]. The recursion feature is able to compress the arbitrary long windows history into a fixed sized hidden layer and then recorded history can help make improved result [36].

Figure 1 is the simplest RNN architecture (Adapted from [37]) and its notations are listed in Table 1. As shown in Fig. 1, RNN is unfolded across time to cover history information. This architecture consists of an input layer w, a hidden layer s with recurrent connections to itself, and an output layer y at the right side. Each layer consists of a certain number of neurons, and the layers are connected with weighs matrices U, W, and V. In language model scenario, the input layer $w(t)$ represents word at the t-th time step encoded with bag-of-words feature. The hidden layer $s(t)$ compresses long history information into itself. The output layer $y(t)$ stands for the probability of next word predicted by $w(t)$ and $s(t-1)$. The dimensions of input vector $w(t)$ and the output vector $y(t)$ are equal to the size of vocabulary. The values of neurons in the hidden layer s and output layer y are computed as follows:

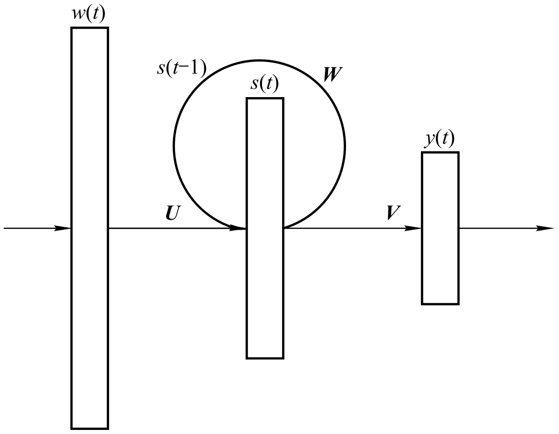

Fig. 1 Architecture of recurrent neural network

$$x(t) = w(t) + s(t-1), \tag{1}$$

where $x(t)$ represents vectors contacting neurons in $w(t)$ and $s(t-1)$.

$$s_j(t) = f\left(\sum_{i=1}^{|V|} x_i(t)U_{ji}\right), \tag{2}$$

$$y_k(t) = g\left(\sum_{j=1}^{|V|} s_j(t)V_{jk}\right), \tag{3}$$

where $f(z)$ is sigmoid activation function like the following one:

$$f(x) = \frac{1}{1+e^{-x}}, \tag{4}$$

and $g(x)$ is multi class function softmax defined as:

$$g(x) = \frac{e^x}{\sum\limits_{k=1}^{|v|} e^k}. \tag{5}$$

Table 1 Notations in recurrent neural network

Symbol	Descriptions
U	Weights metric between input layer and hidden layer
W	Weights metric from hidden layer to hidden layer
V	Weights metric between hidden layer and output layer
$w(t)$	The values at the t-th time step encoded with bag-of-words feature
$s(t)$	The values of hidden layer neuron at time step t
$y(t)$	The values of output layer neuron at time step t

Among equations, the parameters (U, W, V) are learned using standard back-propagation or back-propagation through time and stochastic gradient decent to maximize the likelihood. In the output layer, gradient vector is computed using cross entropy criterion.

2.3 Word embedding

Embedding words into a continuous vector space has a long history in the domain of natural language processing. Bengio et al. proposed a very popular model architecture to construct a neural network language model [35]. In this model, a feed-forward neural network with a linear projection layer and a non-linear hidden layer were used to learn the word vector representation and a statistical language model simultaneously. Neural network language model outperforms the n-gram language model with an elaborated designed smooth function [18]. Analysis indicates that the superior performance is mainly attribute to the better trained word vectors learned by neural network language model. For example, having seen the sentence "Book three tickets from Boston to Bejing" in a training corpus may help the model generalize to the sentence "Book two tickets from HongKong to Washington". That is mainly because that "Boston" and "HongKong", "three" and "two" in the sentences have similarities in both semantic meaning and syntactic structure which in return put them closer in the continuous vector space.

Except feed-forward neural network, many other architectures have also been proposed to train word embedding. Whatever their purpose is to predict a probability of a word given previous ones in a sentence [35,38] or actually to produce a better representation [39–41], all can achieve remarkable performance. Among them, Skip-gram architecture is a widely used mechanism for this task. Skip-gram tries to predict surrounding words based on the current word. More generally, it uses current word as input to a neural network model and tries to predict word within a certain range before or after it. This simple model has been proven to have achieved satisfied performance [42]. Therefore, in this article we utilise Skip-gram [40] to train an high quality embedding efficiently as unsupervised fashion.

More formally, given a sequence of training words $w_1, w_2, w_3, \ldots, w_T$, the objective of the Skip-gram model is to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}[\sum_{j=-k}^{k}\log(p(w_{t+j}|w_t))], \tag{6}$$

where k is the size of the training window (which can be a function of the centre word w_t). The inner summation goes from $-k$ to k to compute the log probability of correctly predicting the word w_{t+j} given the word in the middle w_t. The outer summation goes over all words in the training corpus.

In the Skip-gram model, every word w is associated with two learnable parameter vectors, u_w and v_w. The probability of correctly predicting the word w_i given the word w_j is defined as:

$$p(w_i|w_j) = \frac{\exp({u_{w_i}}^{\mathrm{T}} u_{w_j})}{\sum\limits_{l=1}^{V}\exp({u_l}^{\mathrm{T}} u_{w_j})}, \tag{7}$$

where V is the number of words in the vocabulary.

The word representations computed using neural networks are very interesting because the learned vectors explicitly encode many linguistic regularities and patterns. Somewhat surprisingly, many of these patterns can be represented as linear translations. For example, the result of a vector calculation vec(Madrid) – vec(Spain) + vec(France) is closer to vec(Paris) than to any other word vector [40].

2.4 Semi-supervised learning

Tagging a dataset manually is expensive and time consuming [43]. Furthermore, in sentiment analysis scenario tagging requires strong domain knowledge [44]. On one hand labelled data is less and in short supply, on the other hand unlabelled data is abundant and easy to get on the web [45]. Semi-supervised learning makes it possible to use both labelled and unlabelled data [46]. So far a lot of works have been done on semi-supervised learning and this kind of methods have been widely proven successful [47]. It significantly boosted performance in many NLP related tasks such as name entity recognition [48], syntax parsing [49], machine translation [50], text document classification [51] and etc.

There exist several popular semi-supervised learning algorithms among which Entropy Regularization is a popular implementation which consists in maximizing the following objective function [47]:

$$C(\theta, \lambda; \mathcal{L}_n) = \sum_{i=1}^{l} \log P(y_i|x_i;\theta) + \lambda \sum_{i=l+1}^{n} \sum_{m=1}^{M} P(m|x_i;\theta) \log P(m|x_i;\theta), \tag{8}$$

where θ are the parameters to optimize, l is the number of labelled data, x_i is the labelled instance, y_i is corresponding label, λ is the Lagrange multiplier, m is one possible label value and M is the number of all possible labels.

Another popular method is TSVM [52], which is a support vector machine (SVM) based model benefiting from both labelled and unlabelled data and minimizes the following objective function:

$$\begin{aligned} &\min_{w,\xi_l,\xi_u^*} \{\frac{1}{2} w^{\mathrm{T}} w + C \sum_{l=1}^{n} \xi_l + C^* \sum_{u=1}^{d} \xi_u^*\} \\ &s.t. \ \forall l : y_l(w^{\mathrm{T}} \dot{\phi}(x_l) + b) >= 1 - \xi_l \\ &\forall u : y_u^*(w^{\mathrm{T}} \dot{\phi}(x_u^*) + b) >= 1 - \xi_u^*, \end{aligned} \tag{9}$$

where w is the parameter of the model, C and $C*$ are penalty values for training and transductive examples, d is the number of unlabelled data for transductive learning, n is the number of labelled data, ξ_l and ξ_u are slack variables, x_l and y_l are the labelled instance and its corresponding label, x_u^* and y_u^* are the unlabelled instance and "pseudo label" obtained by transductive learning.

Smith also proposed a contrast estimation on log-linear model to utilize unlabelled data [53]. This method consists in maximizing the following objective function unlabelled instances:

$$\prod_i P(X_i = x_i | x_i \in \mathcal{N}(x_i); \theta), \tag{10}$$

where x_i is an unlabelled instance, $\mathcal{N}(x_i)$ is a set of examples that are perturbations of x_i, and the $\mathcal{N}$ is a mapping which generates a set of perturbations of x. All the above methods allows us to perform semi-supervised learning from both labelled data and unlabelled data.

As to the neural network, previously proposed deep neural networks with traditional back propagation algorithm did not get satisfied performance, partially due to not being initialized properly [19,54]. Traditionally the parameters are initialized as random small weights, which results in a high probability for the parameters to fall into poor local minimums [55]. If they can be initialized properly by pre-training, the performance will be well improved. Besides its contribution to a better initialization, previous work [56] shows that pre-training also acts as a regularizer on the parameters even tough no explicit regularization terms for this effect appeared in the objective function. Suppose parameters are to be chosen from a space S, and S is split into regions R_k that are the basins of attraction of descent optimization procedure which minimizes the training error such that $S \subset \cup R_k$ and $R_i \cap R_j = \emptyset$. Let $v_k = \int 1_{\theta \in R_k} \mathrm{d}\theta$ be the volume associated with R_k, and π_k be the probability that pre-training lands in R_k. We then have $\sum_k \pi_k = 1$. Erhan et al. have further indicated that unsupervised pre-training is equivalent to adding penalty on solutions that are outside the desired parameter space [56], where

$$Regularizer = -\log P(\theta). \tag{11}$$

For pre-trained models the prior P is

$$P_{pre-training}(\theta) = \sum_k 1_{\theta \in R_k} \frac{\pi_k}{v_k}. \tag{12}$$

We can verify that $P_{pre-training}(\theta \in R_k) = \pi_k$, and when π_k is tiny, the penalty is high for $\theta \in R_k$. So pre-training seems to constrain implicitly where the parameters ought to be and thus confine them to optimal positions in the parameter space. As a result, most work around deep neural network are mainly executed in two main procedures, 1) unsupervised pre-training, where the weight or parameters are initialized by layer-wise unsupervised training [57]; 2) fine tuning, where the parameters are further trained globally with labelled data using traditional back propagation algorithm [58].

In this paper, we train separately a set of word embeddings and use it to initialize parameters in the proposed model. The experimental study has shown the improvements with regard to the polarity detection performance.

3 Methodology of SDRNN

In this section, we will present detailed description of our proposed model, the semi-supervised deep recurrent neural network (SDRNN). Firstly, we introduce the architecture of SDRNN accompanying by description of cost function and the reason why we take cross entropy rather than square error root as cost function. Furthermore, we demonstrate how we train the proposed model using back propagation through time algorithm including detailed derivation of error signals in output layer and hidden layer. Since we intend to utilize on abundant unlabelled data which are available on the internet, we also give a characterization on approaches to train word embedding and how it can be deployed to pre-train our model.

3.1 Dual recurrent neural network for sentiment analysis

In this research, a novel discriminative RNN model for sentiment analysis architecture has been proposed, as shown in Fig. 2 and unfolded version in Fig. 3 and its notations are listed in Table 2. Similar to traditional RNN model, it is also unfolded across time to cover long history memory. In sentiment analysis scenario, the whole statement is firstly divided into K parts. As a result the input layer $w(t)$ represents words at the t-th parts of the encoded with bag-of-words feature.

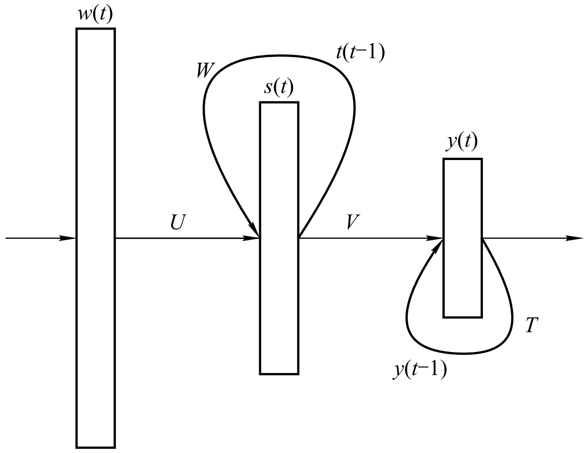

Fig. 2 Architecture of SDRNN

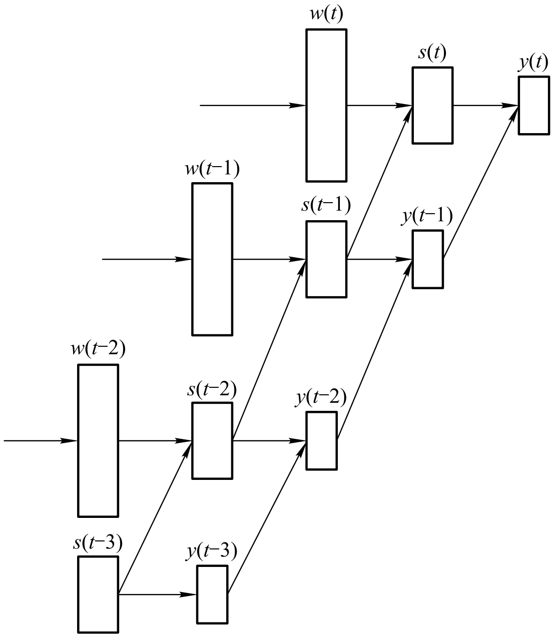

Fig. 3 Dual recurrent neural network unfolded as deep feed-forward neural network

The output layer $y(t)$ stands for the sentiment distribution predicted by $w(t)$. The hidden layer $s(t)$ can compress arbitrary long history information into itself. The dimension of input vector $I(t)$ equals to the size of vocabulary and the output vector $y(t)$ equals to the size of sentiment labels. The values of neurons in the hidden layer s and output layer y are computed as follows:

Table 2 Notations in SDRNN

Symbol	Descriptions
$\mathbf{d}$	Distribution of desired output
$\mathbf{y}$	Distribution of predicted output
d_k	The kth element of desired output
y_k	The kth element of predicted output
$loss(\mathbf{d}, \mathbf{y})$	Loss function between desired output and predicted output
$H(\mathbf{d}, \mathbf{y})$	Cross entropy between desired output and predicted output
o_i	The ith element of output layer

$$x(t) = w(t) + s(t-1), \tag{13}$$

$w(t)$ here stands for the whole words in the t-th parts or segments rather than the t-th words in a sentence:

$$s_j(t) = f\left(\sum_{i=1}^{|V|} x_i(t) U_{ji}\right), \tag{14}$$

where $f(z)$ is sigmoid activation function, you can also use other activation functions:

$$l(t) = s(t) + y(t-1). \tag{15}$$

Intuitively, the sentiment distribution at previous time step would also be possible to contribute to the final polarity detection result. Inspired by this idea, the proposed architecture has been designed to have a recursion from Y to itself, as defined below:

$$y_k(t) = g\left(\sum_{j=1}^{|l|} l_j(t) V_{jk}\right). \tag{16}$$

If we choose mean square error as objective function then error signals can be computed as follows:

$$\begin{aligned} loss(\mathbf{d}, \mathbf{y}) &= \sum_{i=1} (d_k - y_k)^2 \\ \Rightarrow \frac{\partial loss}{\partial o_k} &= (d_k - y_k) \times \frac{\partial y_k}{\partial o_k} \\ &= (d_k - y_k) \times \underbrace{y_k \times (1 - y_k)}_{suboptimal}. \end{aligned} \tag{17}$$

Otherwise, if we take cross entropy [59] as objective function, it is able to get error signals as follows:

$$\begin{aligned} H(\mathbf{d}, \mathbf{y}) &= \sum_i d(X_i) \log(p(X_i)) \\ &= \log p(X_{correct}) = o_{correct} - \log \sum_i e^{o_i} \end{aligned}$$

$$\Rightarrow \frac{\partial H(P,Q)}{\partial o_k} = \delta(k = correct) - \frac{\partial \log \sum_i e^{o_i}}{o_k}$$
$$= \delta(k = correct) - \frac{e^{o_k}}{\sum_i e^{o_i}} = \delta(k = correct) - y_k. \quad (18)$$

In Eq. (17), terms in under-brace may make whole error signal very small since both y_k and 1–y_k are smaller than 1, which makes the model difficult in updating parameters. In this scenario, it is important to use cross entropy rather than mean square error which would leads to suboptimal since the purpose is to minimize entropy and perplexity. As a result, in this work we deploy cross entropy as loss function.

Therefore the objective function is then defined as:

$$\prod_t P(y(t)|x(t)). \quad (19)$$

Error signals in model are computed as follows:

$$\mathbf{e}_o(t) = (\mathbf{d}(t) - \mathbf{y}(t)) \times \frac{K}{t}, \quad (20)$$

where $\mathbf{d}(t)$ is the expected value which represent sentiment label that should be predicted(all encoded with one-hot vector), $\mathbf{y}(t)$ be the predicted sentiment label and K is the number of parts into which we divided the description. Since the t-th part is just a part of the whole statement and do not take the whole information so we discount the error signal to let the model do not trust it completely.

After getting error signals in output layer, what we need to do is propagating it back to hidden layer.

$$\mathbf{e}_h(t) = g_h(\mathbf{e}_o(t)^{\mathrm{T}}\mathbf{V}, \mathbf{t}), \quad (21)$$

where $g_h(x, t)$ is element-wise operation computing values as follow:

$$g_h(x, t) = x. * \mathbf{s}(t). * (1 - \mathbf{s}(t-1)). \quad (22)$$

Other weights are updated using traditional back propagation algorithm:

$$\mathbf{V}(t+1) = \mathbf{V(t)} + \mathbf{s}(t)\mathbf{e}_o(t)\alpha - \mathbf{V}(t)\beta; \quad (23)$$

$$\mathbf{U}(t+1) = \mathbf{U(t)} + \mathbf{w}(t)\mathbf{e}_h(t)\alpha - \mathbf{U}(t)\beta; \quad (24)$$

$$\mathbf{T}(t+1) = \mathbf{T(t)} + \mathbf{y}(t-1)\mathbf{e}_o(t)\alpha - \mathbf{T}(t)\beta; \quad (25)$$

$$\mathbf{W}(t+1) = \mathbf{W(t)} + \mathbf{s}(t-1)\mathbf{e}_h(t)\alpha - \mathbf{W}(t)\beta, \quad (26)$$

where e_o, e_h is respectively error vector of output layer and hidden layer, α and β are learning rate which are manually set. The last term of each equation is regularization in accordance with Occam's razor theory.

However, training algorithms presented above is depicted as back-propagation algorithm. It is out of question since RNNs can be viewed as a normal feed-forward neural network with only one hidden layer assuming final sentiment distribution only depends on previous hidden layer state and previous output layer state. That's to say, the model captured sentiment distribution just based on hidden layer and output layer activation at previous time. In addition, if the model learning context information, it is just happened to be.

Nevertheless, RNNs are designed to model long sequence or segment context information. As such, in this paper we take back-propagation through time, an extension of traditional back-propagation optimization method, which can guarantee the model learn what should be stored in hidden layer and output layer recursion. The procedure goes like this: RNNs with recursion used for k times step can be unfold as a deep feed-forward neural network with k hidden layers assuming weight matrix in these recursion are the same [60].

According to back-propagation through time algorithm [61,62], weights are updated as follows:

$$\mathbf{V}(t+1) = \mathbf{V(t)} + \sum_{\tau=0}^{T} \mathbf{s}(t-\tau)\mathbf{e}_o(t-\tau)\alpha - \mathbf{V}(t)\beta, \quad (27)$$

$$\mathbf{U}(t+1) = \mathbf{U(t)} + \sum_{\tau=0}^{T} \mathbf{w}(t-1-\tau)\mathbf{e}_h(t)\alpha - \mathbf{U}(t)\beta, \quad (28)$$

$$\mathbf{T}(t+1) = \mathbf{T(t)} + \sum_{\tau=0}^{T} \mathbf{y}(t-1-\tau)\mathbf{e}_o(t-\tau)\alpha - \mathbf{T}(t)\beta, \quad (29)$$

$$\mathbf{W}(t+1) = \mathbf{W(t)} + \mathbf{s}(t-1)\mathbf{e}_h(t)\alpha - \mathbf{W}(t)\beta. \quad (30)$$

3.2 Semi-supervised recurrent neural network

So far, the proposed model is only trained by using labelled data. However, as mentioned in earlier section, recurrent neural is a complicated model which has many local optimization. Therefore it is necessary to initialize the model. In this research, we collect movies reviews data from IMDB website and then use these data to conduct training for a set of word embedding.

The reason we train word embedding instead of using popular SENNA embedding is because that our trained word embedding is domain specific which has strong prior towards movie domain. In this article Continuous Skip-gram Model [18] is employed as it has been proven to be efficient and of high quality. Its core point is to use each current word as an input to a log-linear classifier with continuous projection layer, and predict words within a certain range before and after the current word, as shown in Fig. 4 adapted from [18].

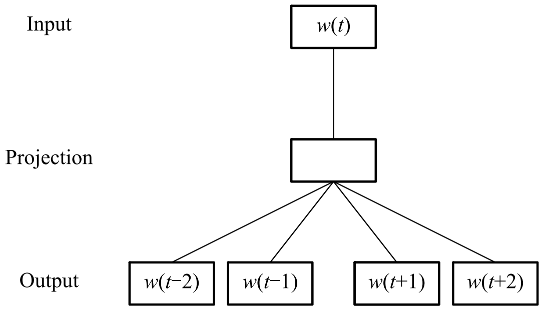

Fig. 4 Architecture of Skip-gram

As such the objective function can be defined as:

$$\prod_{y(t)\in Range(w(t))} P(y(t)|w(t)), \tag{31}$$

where $w(t)$ represents the current word and $Range(w(t))$ stands for words that in a context window over $w(t)$.

After the word embedding has been trained, it can be used to initialize the weights between the input layer and hidden layer. As a result, the network is well initialized and expected get better result when optimized.

To combine the labelled and unlabelled data, the following objective function is employed:

$$\operatorname*{argmin}_{\wedge\in R^k} \Big[-\sum_{i=1}^{l} \log(p(\mathbf{y}_i)|\mathbf{x}_i; \wedge) - \sum_{i=l}^{l+u} \log(p(\mathbf{y}_i^*)|\mathbf{x}_i^*; \wedge)\Big], \tag{32}$$

where the first term is supervised learning part and the second one is unsupervised fashion. $\mathbf{x}_i$ stands for bag-of-words in a description, $\mathbf{y}_i$ stands for sentiment labels while $\mathbf{x}_i^*$ represents words in a sentence and $\mathbf{y}_i^*$ represents words in a context window over $\mathbf{x}_i^*$. The symbol $\wedge$ is parameter in this model, l is the number of labelled data and u is the number of unlabelled data. The overall training procedure is shown in Algorithm 1.

Algorithm 1 Training algorithms of SDRNN model

```
Input:
    Sequence $\langle(x_1, y_1),(x_2, y_2), \ldots, (x_n, y_n)\rangle$
    Learned word embedding using above Skip-gram
    Maximum epoch and learning rate.
Output:
    Learned parameters $\langle U, W, V, T\rangle$ of SDRNN model
Using learned word embedding to initialize parameter $U$
For each training example $(x_i, y_i)$, do feedforward pass obtaining output $p_i$
Computing error signals and updating parameters using Eqs. (27), (28), (29), (30).
Looping until meets maximum epoch.
```

4 Experimental study

4.1 Dataset

For the purpose of comparing the proposed model against other baseline approaches, three well known and widely used datasets are employed in this research, i.e., well balanced dataset movie review and non-balanced dataset MPQA opinion corpus and Customer Review dataset. The description details of the three datasets can be found in Table 3.

Table 3 Dataset summary

Dataset	Number of instance	Positive/Negative
Movie review	10 624	0.5/0.5
MPQA opinion corpus	10 662	0.31/0.69
Customers review	3 772	0.64/0.36

1) **Movie Review**[1]: In this dataset, each items of this dataset is a review on a movie collected from IMDB website and is labelled either positive or negative polarity. This dataset contains 10 662 review snippets and the number of positive and negative sentiments are equal to 5 331 [63].
2) **MPQA Opinion**[2]: The MPQA Opinion Corpus contains news articles from a wide variety of news sources manually annotated for opinions and other private states (i.e., beliefs, emotions, sentiments, speculations, etc.) [64]. This dataset has 10 624 instances, but its distributions of negative and positive are not balanced.
3) **Customer Review**[3]: This dataset is very much alike Movie Review, the difference is that each items is a review on a product (Nokia, Canon, Apex and etc.) and also is labelled either positive or negative. This dataset comprises 3 772 instances, distributions of negative and positive are also not balanced [65].

In this experimental study, all the above data sets are randomly split into training set (70%) and test set (30%).

4.2 Experimental settings and evaluation metrics

Since the proposed SDRNN is an extension of traditional RNN, the main hyper-parameters of SDRNN will be similar to traditional RNN, i.e., hidden layer size, learning rate and regularization penalty factor [37]. Meanwhile, since the input in the proposed SDRNN model is a set of segments, the segment size will be also a decisive factor affecting the overall polarity detection performance. To simplify the validation

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[2] http://mpqa.cs.pitt.edu/
[3] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

in this experimental study, evaluation will be conducted over different hidden layer size and segment size with learning rate equals 0.1 and regularization penalty factor $1e-4$, as these two factors are generally pre-defined in advance [66].

In this research, the performance is mainly evaluated by the effectiveness of classifying a statement into positive and negative. As such the metrics will employ accuracy, which is widely used in information retrieval field [67], as the main measurement, which is defined as below:

$$accuracy = \frac{\sum_{i=1}^{n} 1\{d_i == y_i\}}{\#testset}, \tag{33}$$

where d_i is the desired value on positive or negative and y_i the predicted value.

4.3 Baseline methods

In this research, to show the promising potential of the proposed SDRNN model, some baseline methods are selected to be compared with regard to the performance over different datasets. The four base line models are listed below:

1) **Bag-of-words**: This method simply uses bag-of-words feature and logistic regression to classify a statement into positive or negative. It is a typical and classical binary classification problem [68].
2) **Vote by Lexicon**: In this approach, the polarity of a subjective sentence is decided by voting of each word's prior polarity. This method employs a sentiment lexicon proposed by [69,70] to count the statement's polarity.
3) **Rule-based reversal using a dependency tree**: The polarity of a subjective sentence is deterministically decided basing on rules, by considering the sentiment polarities of dependency sub-trees. The polarity of the dependency sub-tree whose root is the i-th phrase is decided by voting the prior polarity of the i-th phrase and the polarities of the dependency sub-trees whose root nodes are the modifiers of the i-th phrase [68].
4) **Tree-CRF**: This dependency tree based method is widely used for sentiment classification of English subjective sentences using conditional random fields with hidden variables [68].

4.4 Evaluation results

Firstly, we collected one million sentences from IMDB website as unlabelled data. Figure 5 is its entropy variation during training process, which also proved that Skip-gram was able to get well accepted performance since the lowest entropy is almost 100. We have tried different method to train word embedding. Among these method, Skip-gram is the best one in terms of entropy. This is also consistent with Tomas's experiment result [42]. Table 4 future demonstrates what the model learned, words with semantic and syntactic similarity tend to be close to each other in lower dimension space.

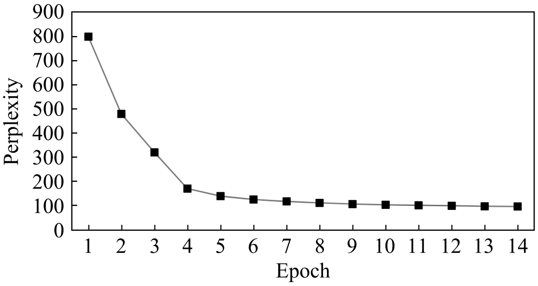

Fig. 5 Entropy variation in unlabelled data

Intuitively, sentiment label distribution of part of description would affect the whole result. Therefore, one of the biggest features of the proposed model is to divide the input into K parts and feed it to the model sequentially. The purpose of adopting this mechanism is to utilize recursion character of RNN. As depicted in Figs. 6(a), 7(a), 8(a) different partition numbers of description would lead to different performance. When K equals 1, the model is just like a simple feed-forward neural network. The performance gets improved with segment partition increasing gradually. In a nutshell, feeding description sequentially boosts the performance than just feeding all word to the model at a time.

Table 4 Word embeddings trained with Skip-gram model using movie domain sentences

Query words	What	A	Bad	Amazing	China	Movie	Fantastic	Fancy
Similar words	why	An	Good	Incredible	Japan	Comedy	Picaresque	Shoeshine
	how	The	Decent	Unnatural	Iran	Cartoon	Captivating	Snappy
	that	Another	Great	Ultimate	Korea	Comic	Climactic	Gruntled
	which	Any	True	Unbelievable	Ethiopia	Television	Frightful	Steppin
	Either	Every	Big	Exceptional	Asia	Film	Heroic	Lovely
	someone	No	Litter	Horrible	Africa	Radio	Voltron	Casual
	you	His	Missing	Awkward	Turkey	Documentary	Calaustrophobic	Trucker
	hillel	This	Possible	Obvious	Thailand	Book	Horrifying	Gaudy
	whether	Its	Bad	Unending	Pakistan	Concert	Horrible	Who

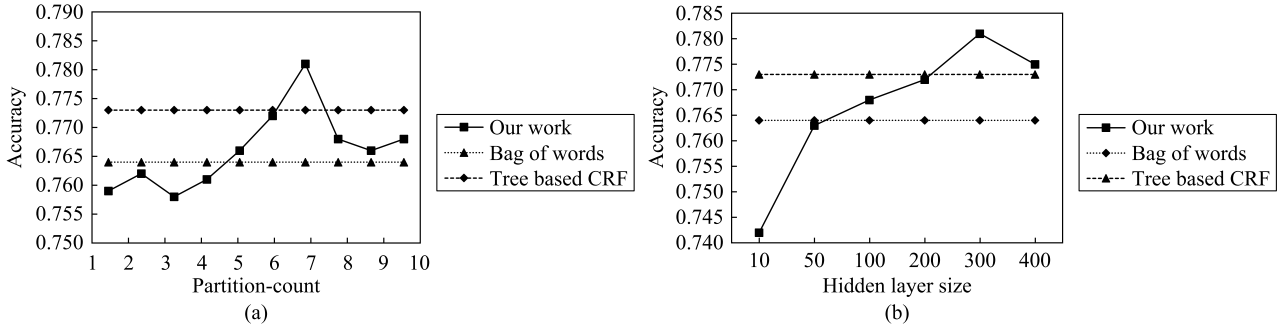

Fig. 6 Accuracy variation over partition count and hidden layer size in movie review dataset. (a) illustrates performance variation over different partition count with hidden layer size being 300; (b) depicts performance variation over different hidden layer size with partition count being 7

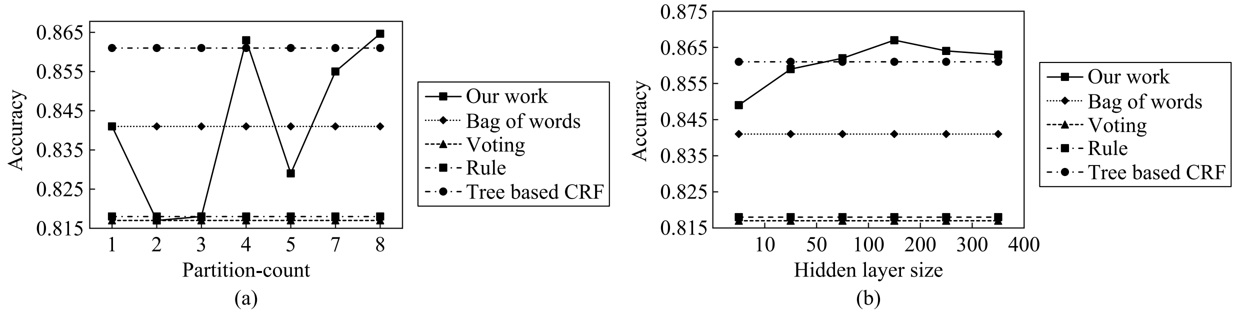

Fig. 7 Accuracy variation over hidden layer size and partition count in MPQA corpus dataset. (a) illustrates performance variation over partition count with hidden layer size being 200; (b) depicts performance variation over different hidden layer size with partition count being 4

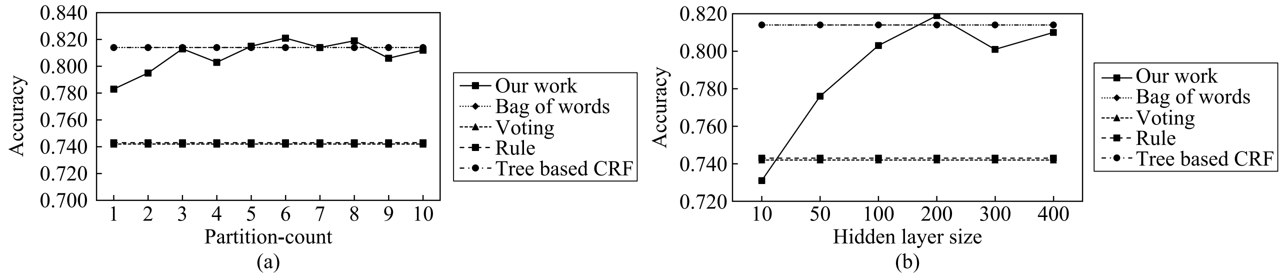

Fig. 8 Accuracy variation over hidden layer size and partition count in customer review corpus dataset. (a) illustrates performance variation over partition count with hidden layer size being 200; (b) depicts performance variation over different hidden layer size with partition count being 6

Figures 6(b), 7(b), 8(b) demonstrates the performance variation over different hidden layer size. Firstly, performance increases with hidden layer size getting larger. After the best one, the performance would decrease. It may partially due to the fact that when hidden layer size becoming larger and larger, parameters of RNN would also squarely increase which in return resulted in over-fitting. One way to tackle this phenomena, which is quite common in neural network models, is to use more date or increase regularization coefficient λ. Since we have already achieved a better number, so we did not tune the pesky parameters though it may get a higher one.

As Fig. 2 shows, this work utilizes dual recursion for hidden layer and output layer respectively. The conducted experimental study has shown that dual recursions from hidden to hidden and output to output can boost performance effectively.

4.5 Discussion

Table 5 is the comparison result of the proposed model against widely used models in the community. In this table, HR means using the hidden to hidden recursion only, DR stands for using recursion both hidden to hidden and output to output recursions, while WEI represents using word embedding trained from above stage to initialize weights between the input layer and the hidden layer. Actually, "SDRNN +HR" represent traditional recurrent neural network. The reason why method "SDRNN+HR" gets poor performance is because neural network is complex and also gets many local optimal point. Nevertheless, when we use semi-supervised fashion we gain well accepted performance though still lower than state-of-the-art system. In addition, we introduced dual recursion to the model from which we obtained state-of-the-art performance though not significant. From the above table, we can draw a conclusion that initialization using word embedding is able to guide the model to global optimization point, and dual recursion can take full advantage of previous sentiment label which results in acceptable result.

Table 5 Accuracy summary over different methods

Method	Movie review	MPQA	Customer review
Bag-of-words	0.764	0.841	0.814
Voting by lexicon	0.631	0.817	0.742
Rule-based reversal	0.629	0.818	0.743
Tree-CRF	0.773	0.861	0.814
SDRNN+HR	0.723	0.829	0.781
SDRNN+WEI+HR	0.767	0.855	0.805
SDRNN+WEI+DR	0.781	0.867	0.821

To further clearly describe what the model can learn, we projected learned embedding, connection weights between input layer and hidden layer, into a two-dimensional space using widely used dimension reduction and visualization tools [71,72], as shown in Fig. 9.

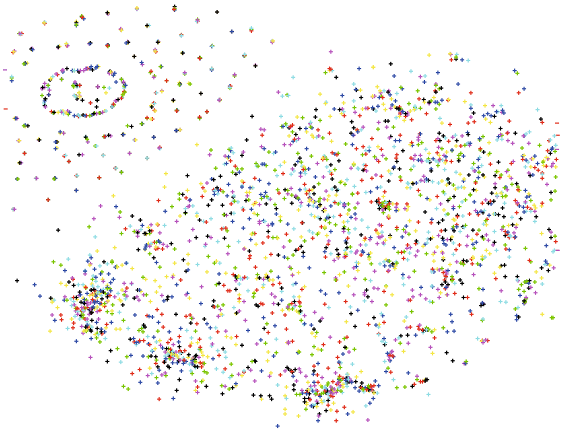

Fig. 9 Embeddings learned by SDRNN model

In Fig. 9, each point represents a word with different colour. As depicted, SDRNN model separates words into two cliques since the output labels are just two categories. Words in the upper left corner of Fig. 9 are those contribute a lot to sentiment labels, such as “wonderful”, “great”, “cool” and etc. Table 6 illustrated those word in detail where the top left corner aggregates words with positive meaning while the bottom right corner are those with negative meaning. In regard to previous sentiment analysis methods, Bag-of-words approach just represents word with one-hot vector which fails to capture semantic and syntactic meanings, and no machine learning at all. Furthermore, voting by lexicon does capture semantic information indeed but building such a lexicon is time consuming and untraceable. While our semi-supervised approach could pull together words with same semantic and syntactic meanings, so long as the training corpus is large enough we can construct an comprehensive lexicon. Moreover, CRF based method is a shallow model compared with SDRNN and cannot learn a highly abstracted representation. However, performance of machine learning is heavily dependent on the choice of data representation [73]. This learned embedding and feedback from past prediction may account for polarity detection performance.

5 Conclusion and future works

In this paper, we have thoroughly investigate the importance of inter-relationship of phrases and words, thereby focusing on from segment’s perspective instead of simply the whole description or each sentences. Afterwards we proposed a discriminative recurrent neural network model to execute sentiment analysis and empirically validated our hypothesis and the results demonstrate that regarding segment as atomic unit would gain better performance. Firstly we utilized a lot of unlabelled data to train a set of word embedding which are used to initialize weights between input layer and hidden layer. In addition, labelled data are also employed to fine tuning the network. In order to use the recursion character, we divide description into K parts and feed them to the proposed model sequentially. Experimental studies are conducted on balanced and non-balanced datasets and the result has proven its improvement on effective polarity detection. It is believed the proposed method will offer researchers in this field insight in neural network based sentiment analysis.

Though the proposed method has shown it potential, some challenges still deserve further efforts and one of them is to investigate whether back propagation through time optimization algorithm can really improve the effectiveness to recurrent neural network [74,75]. To clarify the uncertainty, we may use Newton method or conjugate Newton method to

Table 6 Representative words that are close to each other after supervised learning in Fig. 9

Words in the top left corner	Words in the bottom right corner
World-famous wonderous amazing terrifically	Boring bothersome capricious cheerless
Undisputably top-quality incredible unbelievable	Clamorous complains congestion conspicuous
Alluringly thrilling admiring well-established worthiness	Contradiction deceitful delinquency devilish fake

optimize the network to further investigate the results. In addition, it is also significant to try different training techniques such as auto-encoder [76,77], restricted Boltzmann machines [78] to train word embedding as initialization weights between input and hidden layers.

Acknowledgements This work was partially supported by the National High Technology Research and Development Program of China (2011AA010502), the National Natural Science Foundation of China (Grant No. 61103095), and the Fundamental Research Funds for the Central Universities. We are grateful to Shenzhen Key Laboratory of Data Vitalization (Smart City) for supporting this research.

References

1. Tan C, Lee L, Tang J, Jiang L, Zhou M, Li P. User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2011, 1397–1405
2. Beineke P, Hastie T, Manning C, Vaithyanathan S. Exploring sentiment summarization. In: Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications. 2004
3. Pang B, Lee L, Vaithyanathan S. Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. 2002, 79–86
4. Cardie C, Wiebe J, Wilson T, Litman D J. Combining low-level and summary representations of opinions for multi-perspective question answering. In: Proceedings of New Directions in Question Answering. 2003, 20–27
5. Dave K, Lawrence S, Pennock D M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th International World Wide Web Conference. 2003, 519–528
6. Kim S M, Hovy E H. Automatic identification of pro and con reasons in online reviews. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. 2006
7. Socher R, Pennington J, Huang E H, Ng A Y, Manning C D. Semi-supervised recursive autoencoders for predicting sentiment distributions. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. 2011, 151–161
8. Maas A L, Daly R E, Pham P T, Huang D, Ng A Y, Potts C. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011, 142–150
9. Turney P D. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002, 417–424
10. Li J, Zheng R, Chen H. From fingerprint to writeprint. Communications of the ACM, 2006, 49(4): 76–82
11. Whitelaw C, Garg N, Argamon S. Using appraisal groups for sentiment analysis. In: Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management. 2005, 625–631
12. Hu M, Liu B. Mining and summarizing customer reviews. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004, 168–177
13. Liu X, Zhou M. Sentence-level sentiment analysis via sequence modeling. In: Proceedings of the 2011 International Conference on Applied Informatics and Communication. 2011, 337–343
14. Mikolov T, Kombrink S, Burget L, Cernocký J, Khudanpur S. Extensions of recurrent neural network language model. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing. 2011, 5528–5531
15. Kingsbury B. Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing. 2009, 3761–3764
16. Maas A L, Le Q V, O'Neil T M, Vinyals O, Nguyen P, Ng A Y. Recurrent neural networks for noise reduction in robust ASR. In: Proceedings of the 13th Annual Conference of the International Speech Communication Association. 2012
17. Yao K, Zweig G, Hwang M Y, Shi Y, Yu D. Recurrent neural networks for language understanding. In: Proceedings of the 14th Annual Conference of the International Speech Communication Association. 2013, 2524–2528
18. Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S. Recurrent neural network based language model. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association. 2010, 1045–1048
19. Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527–1554
20. Lafferty J D, McCallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning. 2001, 282–289
21. Elman J L. Finding structure in time. Cognitive science, 1990, 14(2): 179–211
22. Pang B, Lee L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2007, 2(1–2): 1–135
23. Morinaga S, Yamanishi K, Tateishi K, Fukushima T. Mining product reputations on the web. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2002, 341–349
24. Volkova S, Wilson T, Yarowsky D. Exploring sentiment in social media: Bootstrapping subjectivity clues from multilingual twitter streams. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics Volume 2: Short Papers. 2013, 505–510
25. Wilson T, Wiebe J, Hoffmann P. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. Computational Linguistics, 2009, 35(3): 399–433
26. Andreevskaia A, Bergler S. Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses. In: Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics. 2006
27. Higashinaka R, Prasad R, Walker M A. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. 2006
28. Davidov D, Tsur O, Rappoport A. Enhanced sentiment learning using

Twitter hashtags and smileys. In: Proceedings of the 23rd International Conference on Computational Linguistics,. 2010, 241–249

29. Hopfield J J. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 1982, 79(8): 2554–2558
30. Waibel A. Modular construction of time-delay neural networks for speech recognition. Neural computation, 1989, 1(1): 39–46
31. Rowley H A, Baluja S, Kanade T. Neural network-based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(1): 23–38
32. Sanger T D. Optimal unsupervised learning in a single-layer linear feedforward neural network. Neural Networks, 1989, 2(6): 459–473
33. Egmont-Petersen M, de Ridder D, Handels H. Image processing with neural networks—a review. Pattern Recognition, 2002, 35(10): 2279–2301
34. Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504–507
35. Bengio Y, Schwenk H, Senécal J, Morin F, Gauvain J. Neural probabilistic language models. In: Holmes D E, Jain L C, eds. Innovations in Machine Learning. Berlin: Springer, 2006, 137–186
36. Kombrink S, Mikolov T, Karafiát M, Burget L. Recurrent neural network based language modeling in meeting recognition. In: Proceedings of the 12th Annual Conference of the International Speech Communication Association. 2011, 2877–2880
37. Mikolov T. Statistical language models based on neural networks. Dissertation for the Doctoral Degree. Brno: Brno University of Technology, 2012
38. Schwenk H, Gauvain J. Connectionist language modeling for large vocabulary continuous speech recognition. In: Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing. 2002, 765–768
39. Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 160–167
40. Subramanya A, Petrov S, Pereira F C N. Efficient graph-based semi-supervised learning of structured tagging models. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. 2010, 167–176
41. Mnih A, Hinton G E. A scalable hierarchical distributed language model. In: Proceedings of the 22nd Annual Conference on Neural Information Processing Systems. 2008, 1081–1088
42. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013
43. Liu K L, Li W J, Guo M. Emoticon smoothed language models for twitter sentiment analysis. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. 2012.
44. Hu X, Tang J, Gao H, Liu H. Unsupervised sentiment analysis with emotional signals. In: Proceedings of the 22nd International World Wide Web Conference. 2013, 607–618
45. Zhou Z H. Learning with unlabeled data and its application to image retrieval. In: Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence. 2006, 5–10
46. Zhu X, Goldberg A B. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 2009, 3(1): 1–130
47. Chapelle O, Schölkopf B, Zien A, eds. Semi-supervised Learning. Cambridge: MIT Press, 2006
48. Rosenfeld B, Feldman R. Using corpus statistics on entities to improve semi-supervised relation extraction from the web. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. 2007
49. McClosky D, Charniak E, Johnson M. Effective self-training for parsing. In: Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. 2006
50. Ueffing N, Haffari G, Sarkar A. Transductive learning for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. 2007
51. Joachims T. Transductive inference for text classification using support vector machines. In: Proceedings of the the 16th International Conference on Machine Learning. 1999, 200–209
52. Bruzzone L, Chi M, Marconcini M. A novel transductive SVM for semi-supervised classification of remote-sensing images. IEEE Transactions on Geoscience and Remote Sensing, 2006, 44(11-2): 3363–3373
53. Smith N A, Eisner J. Contrastive estimation: Training log-linear models on unlabeled data. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 2005, 354–362
54. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. In: Proceedings of the 20th Annual Conference on Neural Information Processing Systems. 2006, 153–160
55. Erhan D, Bengio Y, Courville A C, Manzagol P A, Vincent P, Bengio S. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research, 2010, 11: 625–660
56. Erhan D, Manzagol P A, Bengio Y, Bengio S, Vincent P. The difficulty of training deep architectures and the effect of unsupervised pretraining. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics. 2009, 153–160
57. Ranzato M, Boureau Y, LeCun Y. Sparse feature learning for deep belief networks. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems. 2007
58. Lee D H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Proceedings of the 2013 ICML Workshop on Challenges in Representation Learning. 2013
59. de Boer P T, Kroese D T, Mannor S, Rubinstein R Y. A tutorial on the cross-entropy method. Annals of Operations Research, 2005, 134(1): 19–67
60. Minsky M, Papert S. Perceptrons-an introduction to computational geometry. Cambridge: MIT Press, 1987
61. Werbos P J. Backpropagation through time: What it does and how to do it. Proceedings of the IEEE, 1990, 78(10): 1550–1560
62. Frinken V, Fischer A, Bunke H. A novel word spotting algorithm using bidirectional long short-term memory neural networks. In: Proceedings of the 4th IAPR TC3 Workshop on Artificial Neural Networks in Pattern Recognition. 2010, 185–196
63. Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics. 2005
64. Wiebe J, Wilson T, Cardie C. Annotating expressions of opinions and emotions in language. Language Resources and Evaluation, 2005, 39(2–3): 165–210
65. Hu M, Liu B. Mining opinion features in customer reviews. In: Pro-

ceedings of the 19th National Conference on Artificial Intelligence and the 16th Conference on Innovative Applications of Artificial Intelligence, 2004, 755–760

66. Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors. In: Polk T A, Seifert C M, eds. Cognitive Modeling. Cambridege: MIT Press, 2002, 213–220
67. Manning C D, Raghavan P, Schütze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press, 2008
68. Nakagawa T, Inui K, Kurohashi S. Dependency tree-based sentiment classification using CRFs with hidden variables. In: Proceedings of the 2010 Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics. 2010, 786–794
69. Stone P J, Dunphy D C, Smith M S. The General Inquirer: A Computer Approach to Content Analysis. Cambridge: MIT Press, 1966
70. Pennebaker J W, Francis M E, Booth R J. Linguistic Inquiry and Word Count: LIWC 2001. Mahway: Lawrence Erlbaum Associates, 2001
71. van der Maaten L, Hinton G E. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9: 2579–2605
72. van der Maaten L, Hinton G E. Visualizing non-metric similarities in multiple maps. Machine Learning, 2012, 87(1): 33–55
73. Bengio Y, Courville A C, Vincent P. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798–1828
74. Martens J, Sutskever I. Training deep and recurrent networks with hessian-free optimization. In: Montavon G, Orr G B, Müller K B, eds. Neural Networks: Tricks of the Trade. 2nd ed. Berlin: Springer, 2012, 479–535
75. Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks. In: Proceedings of the 30th International Conference on Machine Learning. 2013, 1310–1318
76. Cowan J D, Tesauro G, Alspector J, eds. Advances in Neural Information Processing Systems 6. San Francisco: Morgan Kaufmann, 1994
77. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 2010, 11: 3371–3408
78. Teh Y W, Hinton G E. Rate-coded restricted Boltzmann machines for face recognition. In: Proceedings of the 2000 Advances in Neural Information Processing Systems 13. 2000, 908–914

Wenge Rong is an assistant professor at Beihang University, China. He received his PhD from University of Reading, UK in 2010; MS from Queen Mary College, University of London, UK in 2003; and BS from Nanjing University of Science and Technology, China in 1996. He has many years of working experience as a senior software engineer in numerous research projects and commercial software products. His area of research covers data mining, service computing, enterprise modelling, and information management.

Baolin Peng received his BS in computer science from Yantai University, China in 2012. He is pursuing his MS in Beihang University, China. His research interests include machine learning and natural language processing, information retrieval and etc.

Yuanxin Ouyang is an associate professor at Beihang University, China. She received her PhD, and BS from Beihang University, China in 2005, 1997, respectively. Her area of research covers recommendation system, data mining, social networks and service computing.

Chao Li received his BS and PhD degrees in computer science and technology from Beihang University, China in 1996 and 2005, respectively. Now he is an associate professor in the School of Computer Science and Engineering, Beihang University, China. Currently, he is working on data vitalization and computer vision. He is a member of IEEE.

Zhang Xiong is a professor in School of Computer Science of Engineering of Beihang University, China and director of the Advanced Computer Application Research Engineering Center of National Educational Ministry of China. He has published over 100 referred papers in international journals and conference proceedings and won a National Science and Technology Progress Award. His research interests and publications span from smart cities, knowledge management, information systems, intelligent transportation systems and etc.