Learning User and Product Distributed Representations Using a Sequence Model for Sentiment Analysis

Tao Chen and Ruifeng Xu, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
Yulan He, School of Engineering and Applied Science, Aston University, Birmingham, UK
Yunqing Xia, Search Technology Center Asia, Microsoft, Beijing, China
Xuan Wang, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China

Abstract— In product reviews, it is observed that the distribution of polarity ratings over reviews written by different users or evaluated based on different products are often skewed in the real world. As such, incorporating user and product information would be helpful for the task of sentiment classification in relation to reviews. However, existing approaches ignored the temporal nature of reviews posted by users or based on product evaluation. We argue that the temporal relations of reviews might be potentially useful for learning user and product embedding and thus propose employing a sequence model to embed these temporal relations into user and product representations so as to improve document-level sentiment analysis. Specifically, we first learn a distributed representation of each review by a one-dimensional convolutional neural network. Then, taking these representations as pre-trained vectors, we use a recurrent neural network with gated recurrent units to learn distributed representations of users and products. Finally, we feed the user, product and review representations into a machine learning classifier for sentiment classification. Our approach has been evaluated based on three large-scale review datasets from the IMDB and Yelp. Experimental results show that: (1) sequence modeling for the purposes of distributed user and product representation learning can improve the performance of document-level sentiment classification; (2) the proposed approach achieves state-of-the-art results on these benchmark datasets.

1 INTRODUCTION

SENTIMENT analysis aims to detect opinions (or polarities) expressed regarding a given subject or topic from text [1]. With the rapid growth of social media platforms such as microblogging services, social networking sites and short messaging services, people increasingly share their views and opinions online. As such, sentiment analysis has attracted much attention since opinions or sentiments detected from text are potentially useful for downstream applications including recommender systems [2], social network analysis [3], market forecasting [4] and the prediction of political topics [5].

Traditionally, researchers focused on identifying the polarity of text based on language clues extracted from the textual content of reviews [6], [7], [8], [9]. Many recommendation and review sites offer a wealth of information beyond mere ratings, such as opinion holders (hereafter, users) who expressed their views and target entities (hereafter, products) that received the reviews. It is often observed that a lenient user might give higher rating than a critical user even if they post an (almost) identical review, while popular products are likely to receive more praises than less popular ones. The distributions of polarity ratings over reviews written by different users or written for different products are often skewed in the real world [10]. Tang et al. reported that sentiment ratings from the same user (or towards the same product) are more consistent than those from different users (or towards different products) [11]. As such, it motivated researchers to exploit user or product information in sentiment analysis.

Some approaches extracted user, product and review features based on the Bag-of-Words assumption, which were then subsequently incorporated into different machine learning classifiers [12], [13]. Others took advantage of topic models in order to capture the interest distribution of users and the content distribution for products [14].

Recently, deep neural networks have been used for distributed representation learning [15], [16] and have shown promising results in sentiment analysis [17], [18], [19]. It is possible to learn the distributed representation of a user or a product, which essentially captures semantic information contained in the reviews posted by the user or via product evaluation. In such a representation, a user or product is represented as a dense and real-valued vector. In previous studies, user, product and review information is incorporated into a purposely-built neural network model in order to learn distributed representations of users and products for the purposes of document-level sentiment classification [11], [20]. However, existing studies have ignored the temporal order of reviews that a user posted or a product received. We argue that the temporal relations of reviews are potentially helpful for learning user and product embeddings. For example, a product that receives positive reviews initially might be more likely to get positive reviews later on. Sequence models, such as recurrent neural network (RNN), are effective in learning temporal information, and have achieved excellent performance on tasks with a focus on temporal sequences [21].

- *Corresponding Author: Ruifeng Xu (Email: xuruifeng@hitsz.edu.cn).*

In this paper, therefore, we propose a sequence modeling-based neural network approach to embed temporal relations of reviews into the categories of distributed user and product representations (hereafter, **user embeddings** and **product embeddings** for short) learning for the sentiment classification of reviews, in which reviews written by one user or evaluated on one product are considered as a temporal-ordered sequence. Using a document-level composition model, which is a one-dimensional convolutional neural network (1d-CNN), each review is first represented by a **review embedding**. Then, all the reviews by each user are sorted in a temporal order and every review is labeled by its respective rating score. We call such a sequence a **user review sequence**. This can be easily converted into a review embedding sequence by replacing each review text by its corresponding review embedding. **Product review sequences** and **product review embedding sequences** are constructed in a similar way where each sequence contains a set of temporal-ordered reviews for a specific product. We subsequently train a separate RNN with gated recurrent unit (RNN-GRU) [22], which has been shown effective in several sequence-based tasks [23], on the constructed user and product review sequences respectively. Continuous vectors generated in the penultimate layer of the model are regarded as user (or product) embeddings, and they capture important global clues such as user preferences, product qualities and temporal relations of reviews. These embeddings are further integrated with continuous review embeddings in a machine learning classifier for sentiment classification.

We evaluate the effectiveness of our approach empirically using three large-scale review datasets from the IMDB and Yelp. We compare our results with a wide range of baselines including word2vec [15], recursive neural networks [17], paragraph vector [16], user product neural network (UPNN) [11] and a state-of-the-art recommendation algorithm JMARS [14]. Experimental results show that (1) sequence modeling for distributed user and product representation learning can improve the performance of document-level sentiment classification; and (2) the proposed approach achieves state-of-the-art results on these benchmark datasets.

The main contributions of our work are summarized below:

- We formulate user and product representation learning as a sequence modeling problem and successfully employ a RNN-GRU neural network to embed temporal relations of reviews into user and product embeddings.
- We propose a novel approach which combines a neural network model and a traditional machine learning classifier for sentiment classification.
- We conduct extensive experiments on three benchmarking datasets and show that our approach achieves state-of-the-art results for document-level sentiment classification.

The rest of this paper is organized as follows: we discuss related work in Section 2; we then present our approach in Section 3; experimental setup and evaluation results are reported in Section 4; finally, Section 5 concludes the paper and outlines future research directions.

2 RELATED WORKS

Sentiment analysis is the field of study that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions from written languages. These can include reviews, forum discussions, blogs, news, comments or any other types of documents [7]. One of the most prominent tasks in sentiment analysis is sentiment classification.

A large body of work in sentiment classification focused on exploring supervised machine learning approaches using various types of features. Early work [1] trained supervised classifiers including Naive Bayes, maximum entropy, and support vector machines (SVM) on the Bag-of-Words features for document-level sentiment classification. It was found that although supervised classifiers outperformed human-produced baselines, they did not perform well when compared to the traditional topic-based classification task. Turney [6] proposed to use point-wise mutual information (PMI) to calculate the sentiment orientation (positive or negative) of a word or phrase. The average sentiment orientation scores of all words or phrases in a document determines the overall sentiment orientation of the document. Mullen and Collier [24] used PMI, features derived from a knowledge base, topic proximity and syntactic relation features to train SVM. Goldberg and Zhu [25] proposed a graph-based semi-supervised learning algorithm to improve sentiment classification performance using unlabeled reviews. Nakagawa et al. [26] presented a dependency tree based method and conditional random fields (CRF) with hidden variables for sentiment classification of Japanese and English subjective sentences. Bickerstaffe and Zukerman [27] proposed a hierarchical classifier algorithm that accounted for the inter-class similarity of tagged sentiment-bearing texts for document-level multi-class sentiment classification. Kiritchenko et al. [28] presented a supervised statistical text classification approach leveraging a variety of sentiment features derived from high-coverage tweet-specific sentiment lexicons. These lexicons are automatically generated from tweets with sentiment-word hashtags and from tweets with emoticons. Gangemi et al. [29] implemented a model and a tool to detect opinion holders and topics of opinionated sentences by using a heuristic graph mining approach that relied on a machine reader for the semantic web. Xu et al. [30] explored the use of cross lingual resources for opinion mining for resource poor languages by using multi-kernel SVM and transfer learning.

Knowledge-based [31] and linguistic pattern based [32] techniques are also popular because of their accessibility and economy. Popular sources of affect words or multi-word expressions for sentiment classification include the WordNet-Affect [33], SentiWordNet [34] and SenticNet [35].

2.1 Distributed Representation Learning for Sentiment Classification

Feature engineering approaches are labor intensive and time consuming. The distributed representation proposed by Hinton et al. [36] is a low-dimensional real value vector for text representation. This kind of representation is effective for capturing syntactic and semantic relationships. With the rapid development of deep neural networks and parallel computing, automatically learning representation of knowledge attracts much research interest in recent years.

Models for learning distributed representations of knowledge have been proposed at different granularity levels, including word sense level [37], [38], [39], word level [15], [40], [41], [42], phrase level [22], [43], [44], sentence level [17], [19], [45], discourse level [46] and document level [16].

For sentiment classification, Maas and Ng [47] used a probabilistic model of documents to learn semantically focused word vectors, and evaluated word vectors in two sentiment analysis tasks. Tang et al. [48] encoded sentiment information of text together with context of words in a number of neural networks with tailored loss functions to automatically derive sentiment embeddings from massive texts. These sentiment embeddings can be used as word features for sentiment analysis without ad hoc feature engineering. Chen et al. [49] combined a WordNet [50] glosses composition model and a context clustering model to learn word sense embeddings which can be used in sentiment analysis tasks.

Socher et al. [17] proposed a recursive neural tensor network (RNTN) for semantic compositionality over a sentiment treebank which pushed the binary classification accuracy on the Stanford sentiment tree bank from 80% up to 85.4%. Kalchbrenner et al. [18] proposed a dynamic convolutional neural network (DCNN) to handle input sentences with varying length and induced a feature graph over a sentence that is capable of explicitly capturing short and long-range relations. Kim [19] presented two simple CNN models with little hyperparameter tuning which were trained on pre-trained word vectors for sentence-level classification tasks. Tang et al. [51] introduced a gated recursive neural network to learn continuous document representations for sentiment classification.

2.2 User and Product Modeling for Sentiment Classification

In recent years, there has been growing interests in incorporating the user and product information for sentiment analysis of product reviews. Seroussi et al. [13] presented a nearest-neighbor collaborative approach for training user-specific classifiers whose outputs were subsequently combined with user similarity measurement for sentiment inference from text. Li et al. [12] used the user, product and review features as a three-dimension tensor, and employed tensor factorization techniques to alleviate the data sparsity problem. Gao et al. [52] referred to user- or product-specific sentiment polarity biases as user leniency and product popularity, respectively. They built a model that automatically computed user leniency and product popularity for sentiment classification. Diao et al. [14] proposed a probabilistic model based on collaborative filtering and topic modeling to capture user and product features for sentiment classification. Li et al. [53] incorporated textual topic and user-word factors with supervised topic modeling. Zhang et al. [54] formalized the phrase level sentiment polarity labeling problem in a convex optimization framework, and designed iterative updating algorithms for leveraging review-level sentiment classification techniques to boost the performance of phase-level sentiment polarity labeling. Tang et al. [11], [20] incorporated user, product and review information into a purposely-built neural network model to learn distributed representations of users and products for document-level sentiment classification.

2.3 Sequence Modeling for Sentiment Analysis

Although sentiment classification has been mostly approached as a binary or multi-class classification problem, some studies considered sentiment analysis as modeling of sentiment flow throughout a document, using sequence modeling approaches for sentiment detection. Mao and Lebanon [55] developed a variant of CRF for sentiment flow prediction. Liu and Zhou [56] decomposed a sentence into a series of sub-sequences using a hidden CRF, and determined the sentence-level polarity by classifying within sub-sequences and by fusing the obtained sub-sequence polarities.

Sequence modeling has often been used for fine-grained sentiment analysis. It aims to detect the subjective expressions in a text and to characterize their intensity and sentiment as well as to identify the opinion holder and the target, or topic, of the opinion [57]. Johansson and Moschitti [58] demonstrated relational features derived from dependency-syntactic and semantic role structures are useful for sentiment analysis. Yang and Cardie [59] proposed a semi-CRF-based approach relaxing the Markovian assumption inherent to CRFs and operated at the phrase level rather than the token level, allowing the incorporation of phrase-level features. Irsoy and Cardie [60] applied deep RNNs to the task of opinion expression extraction formulated as a token-level sequence-labeling task.

In this paper, we proposed to learn user and product embeddings from the temporal ordered reviews written by a user or evaluated on a product respectively using a variant of RNN as a sequence learning model combining user, product and review content information.

3 METHODOLOGY

In this section, we present our approach for learning user and product distributed representations using a sequence model for sentiment analysis. An overview of the approach is shown in Fig. 1. We first describe the document composition model which produces the distributed representation of each review document (Section 3.1). Afterwards, we introduce the sequence model for embedding temporal relations of reviews into user and product representations learning (Section 3.2). Finally, we describe the sentiment classification model which encodes user, product and review information (Section 3.3).

3.1 Modeling Reviews with Multi Filter 1d-CNN

We describe our approach for learning review embeddings from review content and its rating with a document composition model (multi filter 1d-CNN).

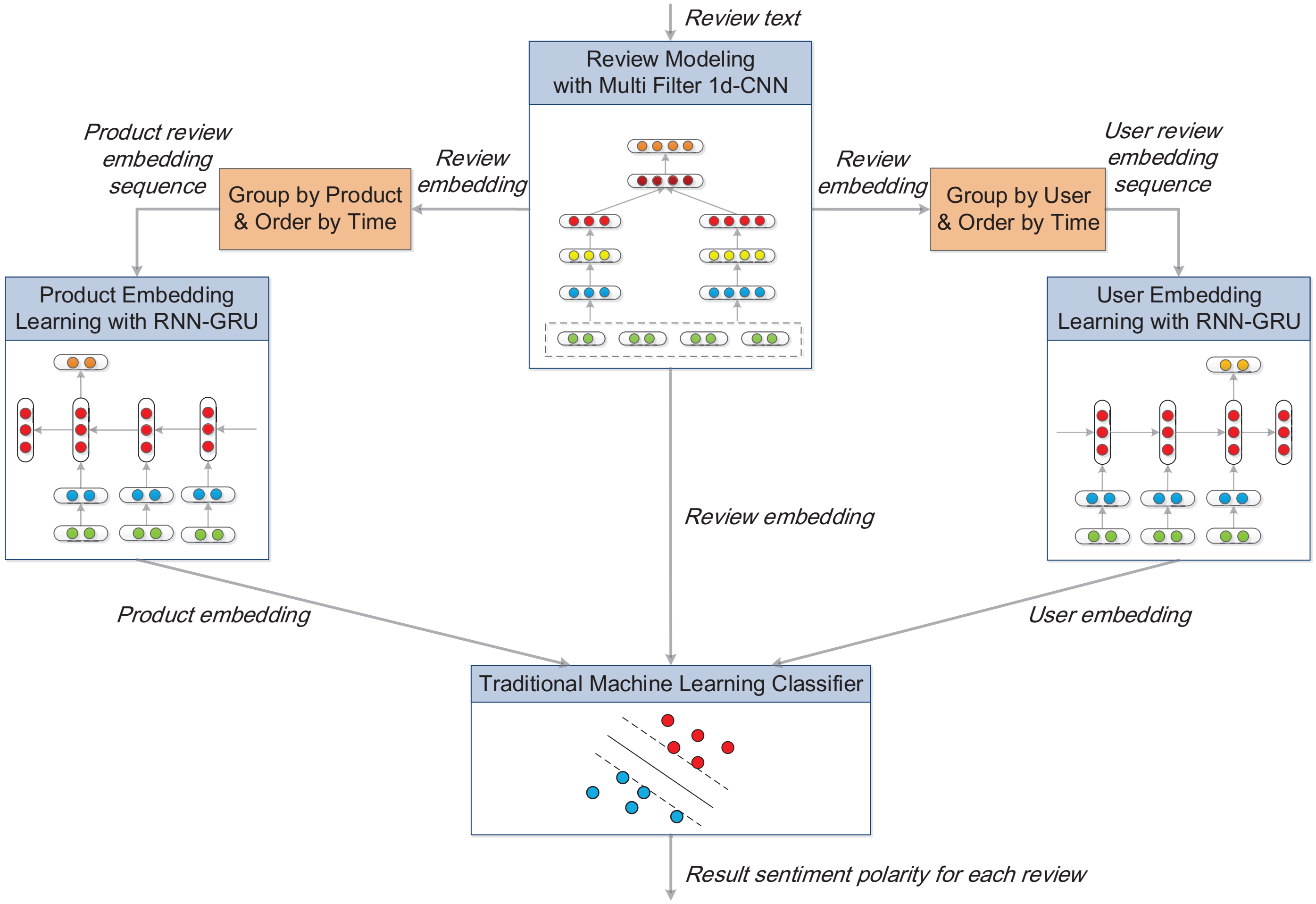
Review text
Review Modeling with Multi Filter 1d-CNN
Product review embedding sequence
Review embedding
Group by Product & Order by Time
Review embedding
Group by User & Order by Time
User review embedding sequence
Product Embedding Learning with RNN-GRU
User Embedding Learning with RNN-GRU
Review embedding
Product embedding
User embedding
Traditional Machine Learning Classifier
Result sentiment polarity for each review

Fig. 1. Framework of our approach.

3.1.1 Training Objective

For reviews with respect to $1 - K$ rating scales (sentiment strength scores), the training objective of the document composition model is to minimize the ranking loss below:

$$\sum_{d\in T} \max\{0, 1 - \mathrm{g}(d) + \mathrm{g}(d')\} \qquad (1)$$

where d is a review document in the training set T with a certain rating from 1 to K (positive sample); d' is another review in T with a rating different from d (negative sample); $\mathrm{g}(\cdot)$ is a scoring function which represents the whole neural network architecture without the last classification layer; $\mathrm{g}(d)$ and $\mathrm{g}(d')$ are the score of positive and negative sample, respectively. For each review in the training set, we expect $\mathrm{g}(d)$ to be approximating 1, $\mathrm{g}(d')$ to be approximating 0, and $\mathrm{g}(d)$ to be larger than $\mathrm{g}(d')$ by a margin of 1 after training the neural network.

3.1.2 Neural Network Architecture

The multi-filter 1d-CNN, with its architecture shown in Fig. 2, is used to learn review embeddings. It takes reviews of varying lengths as input and produces fixed-length vectors as output.

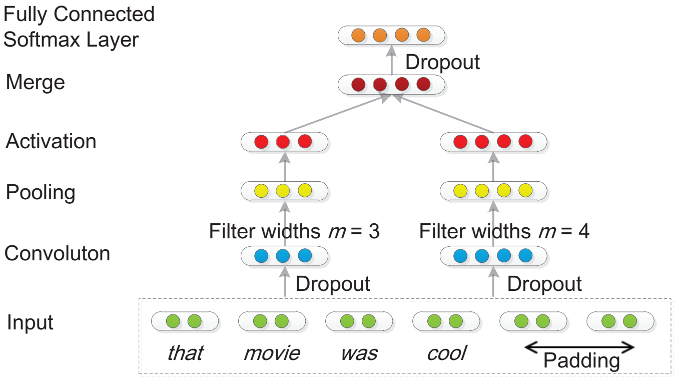

Fig. 2. An illustration of 1d-CNN with two filters for an example review.

Before training, a distributed representation for each word (often referred to as a **word embedding** [41]) in all reviews is generated. It can be initialized randomly or taken from other pre-trained word embeddings. All the word embeddings are stacked in a matrix $M \in \mathbb{R}^{d\times|V|}$, where d is the dimension of word embedding and $|V|$ is the size of word vocabulary. In the input layer, embeddings of words in the current training review are taken from M. In order to handle input reviews of varying lengths, the maximum length of reviews that the network handles is set to m. Shorter reviews are padded with zero vectors. Then, dropout regularization, introduced by Hinton et al. [61], is used to control over-fitting.

In the convolution layer, a filter moves on the word embeddings to perform one-dimensional convolution. The idea behind the one-dimensional convolution is to take the dot product of the filter vector $w \in \mathbb{R}^{nd}$ with each n-gram in the review r to obtain another sequence c. As the filter moves on, many sequences are generated. We call them feature sequences. The i-th feature sequence $c_i \in \mathbb{R}$ is generated as follows:

$$c_i = \mathrm{f}(w \cdot r_{i:i+n-1} + b) \tag{2}$$

where n is the size of filter; $b \in \mathbb{R}$ is a bias term and $\mathrm{f}(\cdot)$ is a point-wise non-linear activation function, such as the hyperbolic tangent (tanh), sigmoid or rectified linear units (ReLU); $r_{i:i+n-1}$ refers to word embeddings of r in current filter window. Many features are combined into a feature map, $F_j \in \mathbb{R}^{n+m-1}$, as defined below:

$$F_j = [c_1, c_2, \cdots, c_{n+m-1}]. \tag{3}$$

In the pooling layer, a max-overtime pooling operation [62], which forces the network to capture the most useful local features produced by the convolutional layers, is applied over F_j. We further add activation functions to incorporate element-wise nonlinearity. The outputs of multiple filters are concatenated in the merge layer. After another dropout process, a fully connected softmax layer output the probability distribution over labels from multiple classes.

3.1.3 Training

The softmax function is an activation function often implemented at the final layer of a network used for classification [63]. It predicts the probability distribution over classes given the input. In this K-classes task, given an input review document d and its representation vector v_d, a conditional tag probability $p(y = i|v_d)$ by applying a softmax operation is defined as follows:

$$p(y = i|v_d) = \frac{e^{{v_d}^T w_i + b_i}}{\sum_{k=1}^{K} e^{{v_d}^T w_k + b_k}} \tag{4}$$

where $i \in 1, \ldots, K$, w and b are the parameters of the merge layer.

The network is trained to minimize the mean of the negative log-likelihood of the prediction of this model under a given target distribution, which is computed as follows:

$$\mathrm{mean}\{\forall i \in i, \ldots, K| - logp(y = i|v_d)\}. \tag{5}$$

The training error propagates back to fine-tune the parameters of the networks and the input word embeddings in each mini-batch, and Adagrad [64] is performed as the gradient descent optimization method. The vectors generated in the merge layer are regarded as review embeddings which capture the semantic features of the input reviews, to some degree.

3.2 Learning User and Product Embeddings from Temporal-ordered Reviews with Sequence Modeling

In this section, we explain how the temporal information about a user or a product, i.e. temporal order of the reviews written by one user or evaluated on one product, is captured in our approach. As shown in Fig. 1, the obtained review embeddings are grouped by the identical user or the identical product, respectively. In each group, review embeddings with corresponding ratings (labels) are treated as a temporal-ordered sequence ordered by their posted time to create a user review embedding sequence and a product review embedding sequence, respectively. Then, these two sequences are fed to a sequence model to learn user embeddings and product embeddings, respectively. We present here a RNN-GRU neural network for sequence modeling.

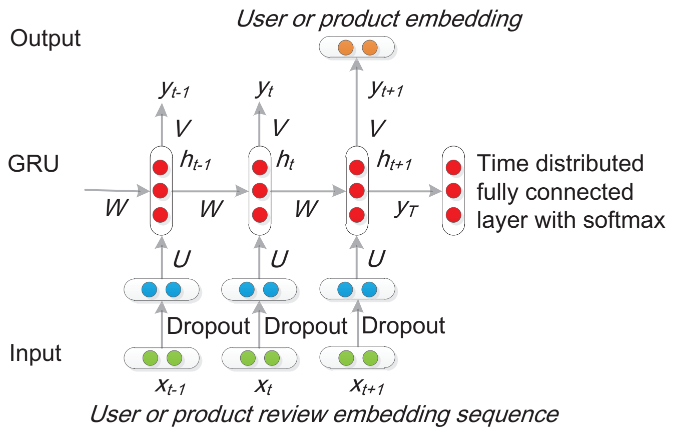

Fig. 3. An illustration of RNN-GRU for user and product review sequence modeling.

3.2.1 Recurrent Neural Network with Gated Recurrent Unit

As illustrated in Fig. 3, a RNN with GRU for learning user or product embeddings is a neural network that takes a user or product review embedding sequence $x = (x_1, \ldots, x_T)$ as input, consisting of a hidden GRU h and an optional output y. T is the last time step. Time refers to the idea that a sequence has a notion of order, e.g., a review sequence ordered by posted time.

At each time step t, the hidden state h_t of the RNN is computed based on the previous hidden state h_{t-1} and the input at the current step x_t:

$$h_t = \mathrm{f}(Ux_t + Wh_{t-1}) \tag{6}$$

where U and W are parameter matrices of the network, and $\mathrm{f}(\cdot)$ is a non-linear activation function. It may be as simple as an element-wise logistic sigmoid function and as complex as a long short-term memory (LSTM) unit [65] or GRU [22].

Both LSTM and GRU are RNN architectures explicitly designed to deal with vanishing gradients problem [66] and efficiently learn long-range dependencies through a gating mechanism. A GRU has two gates, a reset gate l, and an update gate z. They are computed as follows:

$$l = \delta(U^l x_t + W^l h_{t-1}) \tag{7}$$

$$z = \delta(U^z x_t + W^z h_{t-1}) \tag{8}$$

where $\delta(\cdot)$ is the logistic sigmoid function, U^l, W^l, U^z and W^z are weight matrices which are learned in the network training process. The reset gate controls when the hidden state ignores the previous hidden state and resets with the current input only. The update gate controls how much information from the previous hidden state will carry over to the current hidden state [22].

The actual activation of the proposed unit h_t is computed as follows:

$$h_t = (1 - z) \odot \hat{h} + z \odot h_{t-1} \tag{9}$$

$$\hat{h} = \tanh(U^{\hat{h}} x_t + W^{\hat{h}} (h_{t-1} \odot l)). \tag{10}$$

The output at step t is computed as follows:

$$y_t = \text{softmax}(V h_t) \tag{11}$$

where V is another weight parameter of the network and softmax is an activation function as described in section 3.1.3. The output vector at the last time step y_T is regarded as the user or product embedding according to the type of the input review sequence.

For user embedding training, input sequence x_t is set to the t-th review embedding in a user review embedding sequence, and for product embedding training, it is set to the t-th review embedding in a product review embedding sequence. h_0 can be initialized to a random vector of small values, h_{t+1} can be initialized to a copy of h_t. A standard back-propagation algorithm with Adam stochastic optimization method is used to train the network. After each training epoch, the network is tested on validation data. The log-likelihood of validation data is computed for convergence detection.

3.3 Sentiment Classification

Up to now, we have embedded variable-length reviews, user review sequences and product review sequences in a fixed-dimensional space which can capture semantic relation in review content, enabling more efficient similarity comparison. A major advantage of embeddings in a fixed-dimensional space is that a wide variety of machine learning algorithms are then applicable.

We concatenate the three embeddings into a single embedding for each review in the training set. Reviews written by the same user or evaluated on the same product use the same user embedding and product embedding, respectively. Then, the concatenated embeddings are used to train a SVM [67] for sentiment classification.

We have also tried to concatenate the three embeddings and use a neural network with a softmax function at the final layer for classification. Experiment results show that training SVMs with the concatenated embeddings achieves better performance than using a neural framework model (detailed in section 4.3.3).

4 EXPERIMENT

We conduct experiments to evaluate the performance of the proposed approach for document-level sentiment classification on three datasets. In this section, we describe the experimental setup and baseline methods followed by the discussion of results.

4.1 Experimental Setup

The three large-scale datasets[1] include one movie review dataset from IMDB developed by Diao et al. [14], and two restaurant review datasets from the Yelp Dataset Challenge[2] in 2013 and 2014 developed by Tang et al. [11]. The statistics of the three datasets are summarized in Table 1. The rating scale used in the IMDB dataset is 1-10, whereas the rating scale used in Yelp 2013 and 2014 datasets is 1-5. We train the rating predictor on the training set, tune parameters on the development set and evaluate on the test set.

We conduct sentiment classification on these three datasets. Following Tang et al.'s work [11], we use accuracy as the evaluation metric to measure the overall sentiment classification performance, and use mean absolute error (MAE) and root mean squared error (RMSE) as the evaluation metrics to measure the divergences between predicted and ground truth sentiment ratings. MAE and RMSE are computed as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |gold_i - predicted_i| \tag{12}$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (gold_i - predicted_i)^2} \tag{13}$$

where $gold_i$ is the true rating for the i-th review, $predicted_i$ is the predicted rating and N is the total number of reviews in our test set. Smaller values indicate more accurate prediction, and hence a better model.

For training CNN, we use: ReLU as activation function, filter windows of 3, 4, 5 with 100 feature maps each, Adadelta decay parameter of 0.95, dropout rate of 0.5, the size of initial word vectors of 300. For training RNN-GRU, we use: ReLU as activation function, dropout rate of 0.25, time distributed fully connected layer with softmax, Adam as stochastic optimization method, categorical cross-entropy as loss function, 300 hidden units.

4.2 Baseline Methods

We benchmark the following baseline methods as having been previously used in [11] for document-level sentiment classification: Trigram, TextFeature, Trigram+UPF, TextFeature+UPF, JMARS, AvgWordvec+SVM, SSWE+SVM, Paragraph Vector, RNTN+Recurrent, UPNN (no UP), and UPNN (full).

Trigram, TextFeature, Trigram+UPF, and TextFeature+UPF are hand-crafted text features based methods. In Trigram, unigrams, bigrams and trigrams are used as features to train a SVM classifier. In TextFeature, word/character n-grams, sentiment lexicon features, and negation features are used [28]. In Trigram+UPF and TextFeature+UPF, user-leniency features [52] and corresponding product features from training data are concatenated with the features in Trigram and TextFeature, respectively.

JMARS is a state-of-the-art recommendation algorithm proposed by Diao et al. [14], in which user and aspects

[1] Available at: http://ir.hit.edu.cn/~dytang/paper/acl2015/dataset.7z

[2] http://www.yelp.com/dataset_challenge

TABLE 1
Statistics of IMDB, Yelp 2014 and Yelp 2013 datasets. #review, #user and #prod denote the number of reviews, users and products, respectively.

DATASET	TRAINING SET			DEVELOPMENT SET			TEST SET			CLASSES
	#REVIEW	#USER	#PROD	#REVIEW	#USER	#PROD	#REVIEW	#USER	#PROD	
IMDB	67,426	1,310	1,635	8,381	1,310	1,635	9,112	1,310	1,635	10
YELP 2013	62,522	1,631	1,633	7,773	1,631	1,633	8,671	1,631	1,633	5
YELP 2014	183,019	4,818	4,194	22,745	4,818	4,194	25,399	4,818	4,194	5

features of a review are used with collaborative filtering and topic modeling.

AvgWordvec+SVM, and SSWE+SVM are two distributed word representation based methods. In AvgWordvec+SVM, word embeddings are learned by word2vec [15], and the mean vector of each word in a document is used as document representation. In SSWE+SVM, sentiment-specific word embeddings (SSWE) [68] are learned, and document representation are generated by max/min/average pooling. Both of them train SVM classifiers for document sentiment classification.

Paragraph Vector, RNTN+Recurrent, UPNN (no UP), and UPNN (full) are distributed document representation based methods. Paragraph Vector is an unsupervised framework proposed by Le and Mikolov [16]. In RNTN+Recurrent, sentence representations are learned by recursive neural tensor network (RNTN) [17], and document representations are composed with RNN. UPNN uses a CNN to compose reviews written by the same user or written on the same product for sentiment classification of documents [11]. In this method, the temporal order of reviews is ignored. UPNN (no UP) uses review content only without considering user and product information and UPNN (full) uses review content, user and product information. Tang et al. report that UPNN (full) achieves state-of-the-art performances on IMDB, Yelp 2013 and 2014 datasets in their work [11].

4.3 Results

4.3.1 Overall Comparison

Table 2 shows the results achieved on the IMDB, Yelp 2013 and Yelp 2014 datasets. The best results are highlighted in bold face. The methods marked with a star use both user and product information in addition to review content, while others only use review texts. The results of the top 11 approaches have been previously reported in [11]. The bottom two approaches are ours. We present the results obtained using our approach without user and product information, denoted as our approach (no UP), or with review content, user and product information taken into account, denoted as our approach (full).

It is observed that for the four text feature based methods, Trigram gives comparable performance to TextFeature which relies on hand-crafted features. Incorporating user and product features improves the classification performance of Trigram and TextFeature on all the three metrics and across all the three datasets. Topic model based method, JMARS, performs similarly compared to text feature based methods on the IMDB dataset, but gives worse results on the other two datasets.

Distributed word representation based methods and distributed document representation based methods can automatically generate features for classifier training. However, two word embedding based methods and Paragraph Vector perform worse than hand-crafted features based methods. RNTN+Recurrent and UPNN (no UP) give mixed results compared to hand-crafted features based methods. UPNN (full) outperforms all the other baselines by using CNN composition and taking into account of all three types of information, including review content, users and products.

Our sequence modeling based approach outperforms all the baselines by a large margin. Without using user and product information, our approach (no UP) gives relative improvements of 15.6% in accuracy, 12.3% in MAE, and 8.7% in RMSE compared to UPNN (full) on the IMDB dataset. On Yelp 2013 and 2014 datasets, the ranges of relative improvements are 6.7–8.1% in accuracy, 13.0–14.8% in MAE, and 10.8–12.2% in RMSE, respectively.

With the user and product information, our approach (full) improves upon UPNN (full) by 12.2% in accuracy, 13.1% in MAE, and 9.4% in RMSE on the IMDB dataset. On Yelp 2013 and 2014 datasets, the ranges of relative improvements are 5.1–7.2% in accuracy, 11.9–14.9% in MAE, and 9.9–11.5% in RMSE, respectively. It shows that the use of sequence modeling for distributed user and product representation learning is effective in improving the performance of document-level sentiment analysis.

We also observe that with user and product information, our approach (full) gives superior performance compared to our approach (no UP) on all the three datasets, with the relative improvements ranging between 2.4–4.3% in accuracy, 3.9–5.8% in MAE and 2.3–2.7% in RMSE, respectively. We apply two-sample t-test on the experimental results of these two approaches, the result indicates statistical significance on all the three datasets ($p < 0.05$). All these results show that incorporating user and product embeddings can boost the performance compared to the models without using them.

4.3.2 Review Sequence Approach (RNN-GRU) vs. Unordered Set Approach (CNN)

We design a CNN version of our approach which ignores the temporal order of the reviews when learning user or product embeddings. For user embedding learning, we aggregate all the reviews written by the same user and concatenate them into a long review with temporal order ignored before feeding it to the CNN. We do it similarly for product embedding learning by aggregating all the reviews about the same product without considering their temporal relation. Other components of our proposed approach, such as learning review embeddings and concatenating review

TABLE 2
Experimental result of sentiment classification on IMDB, Yelp 2013 and Yelp 2014 datasets. For accuracy, higher is better. For MAE and RMSE, lower is better.

	IMDB			YELP 2013			YELP 2014		
	ACCURACY	MAE	RMSE	ACCURACY	MAE	RMSE	ACCURACY	MAE	RMSE
TRIGRAM	0.399	1.147	1.783	0.569	0.513	0.814	0.577	0.487	0.804
TEXTFEATURE	0.402	1.134	1.793	0.556	0.520	0.845	0.572	0.490	0.800
TRIGRAM+UPF*	0.404	1.132	1.764	0.570	0.491	0.803	0.576	0.471	0.789
TEXTFEATURE+UPF*	0.402	1.129	1.774	0.561	0.509	0.822	0.579	0.476	0.791
JMARS*	N/A	1.285	1.773	N/A	0.699	0.985	N/A	0.710	0.999
AVGWORDVEC+SVM	0.304	1.361	1.985	0.526	0.568	0.898	0.530	0.562	0.893
SSWE+SVM	0.312	1.347	1.973	0.549	0.529	0.849	0.557	0.523	0.851
PARAGRAPH VECTOR	0.341	1.211	1.814	0.554	0.515	0.832	0.564	0.496	0.802
RNTN+RECURRENT	0.400	1.133	1.764	0.574	0.489	0.804	0.582	0.478	0.821
UPNN (NO UP)	0.405	1.030	1.629	0.577	0.485	0.812	0.585	0.483	0.808
UPNN (FULL)*	0.435	0.979	1.602	0.596	0.464	0.784	0.608	0.447	0.764
OUR APPROACH (NO UP)	0.468	0.903	1.487	0.624	0.413	0.713	0.624	0.410	0.704
OUR APPROACH (FULL)*	**0.488**	**0.851**	**1.451**	**0.639**	**0.395**	**0.694**	**0.639**	**0.394**	**0.688**

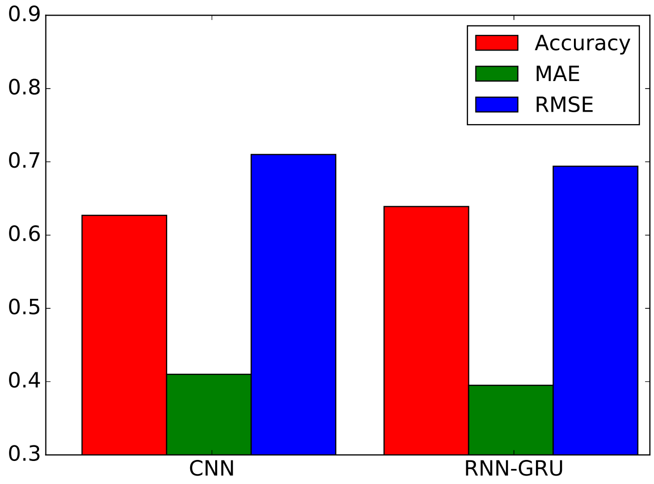

Fig. 4. Experimental results of review sequence approach (RNN-GRU) and unordered set approach (CNN) on the Yelp 2013 dataset.

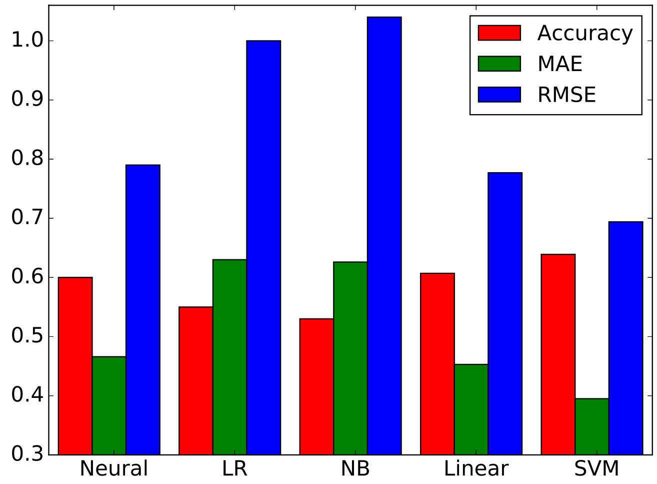

Fig. 5. Experimental results of traditional classifiers vs. a pure neural framework on the Yelp 2013 dataset.

content, user and product embeddings for SVM training, remain the same. We call this variant unordered set approach since all the reviews of the same user or the same product are considered as an unordered set. Our original approach using RNN-GRU for user and product embedding learning is termed as review sequence approach.

Fig. 4 shows the comparison results on the Yelp 2013 dataset. It can be observed that learning user and product embeddings from temporal ordered reviews gives better performance compared to learning from unordered set of reviews. This shows that the temporal relations indeed characterize users and products better.

4.3.3 *Comparison of Traditional Classifiers with a Pure Neural Framework*

There are two main steps in our approach: firstly, learning user, product and review embeddings; secondly, feeding these embeddings into SVM training. We also design a pure neural version of our approach, in which SVM is replaced by a fully connected layer with the softmax function and hence sentiment classification can be performed in the process of representation learning.

We present the experimental results of this pure neural framework and our approach using four different classifiers in Fig. 5, where Neural indicates this pure neural framework. LR, NB and Linear indicate our approach with SVM replaced by logistic regression [69], Naive Bayes, and LIBLINEAR [70], respectively.

It is observed that neural framework outperforms logistic regression and Naive Bayes, but gives worse results compared to LIBLINEAR and SVM on all the three metrics. The best performance is obtained using SVM, which outperforms the neural framework by a large margin. This shows that for the datasets experimented here, it is better to separate representation learning from sentiment classifier training.

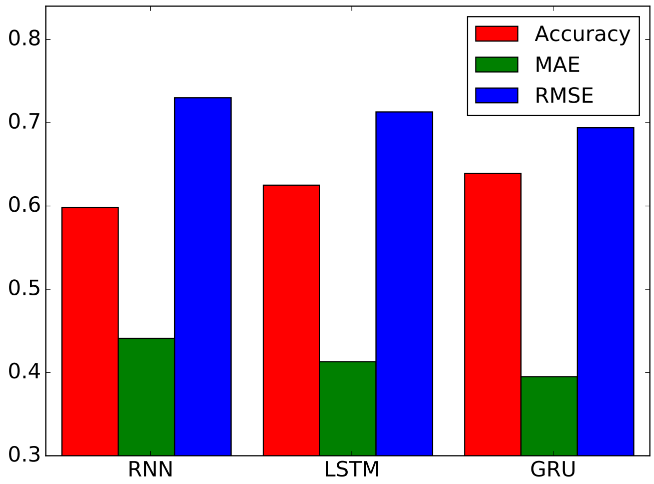

Fig. 6. Experimental results of different sequence models on the Yelp 2013 dataset.

4.3.4 *Comparison with Different Sequence Models*

We have also experimented with different sequence models, including RNN and LSTM, for learning user and product representations. Fig. 6 shows the results of using RNN, LSTM and GRU as a sequence model on the Yelp 2013 dataset. It can be observed that GRU outperforms LSTM, which in turn gives better performance compared to RNN on all the three metrics. It is because both GRU and LSTM, having more complicated hidden units, offer better composition capability than RNN. GRU has fewer parameters to train compared to LSTM and thus generalizes better than LSTM.

5 Conclusion

This paper has presented a sequence modeling based neural network approach for document-level sentiment analysis. The approach employs RNN-GRU to learn user and product embeddings from the temporal ordered review documents. These embeddings, together with review embeddings learned by a CNN, are used to train SVMs for sentiment classification. We have conducted extensive experiments on three review datasets using three evaluation metrics. Empirical results show that our approach achieves the state-of-the-art performance on all these datasets. We have found that (1) modeling reviews as sequences rather than unordered sets boost the performance of user and product representation composition; (2) concatenating review, user and product embeddings for training SVMs for sentiment classification gives superior results compared to a pure neural framework and beats the best results reported so far. Evaluations on three large-scale datasets show that the proposed method performs better than several strong baseline methods which regard reviews as unordered set.

In future work, we plan to explore other sequence learning model, such as bidirectional RNN, bidirectional LSTM and gated feedback RNN for sentiment analysis. We will also explore other methods in learning user and product embeddings and investigate the feasibility of using these embeddings for a wide range of tasks such as product recommendation and product sales prediction.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61370165, 61203378), National 863 Program of China 2015AA015405, Shenzhen Development and Reform Commission Grant No. [2014]1507, Shenzhen Peacock Plan Research Grant KQCX20140521144507925, Shenzhen Foundational Research Funding JCYJ20150625142543470 and Baidu Collaborate Research Funding.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2002, pp. 79–86.

[2] R. L. Rosa, D. Z. Rodríguez, and G. Bressan, "Music recommendation system based on user's sentiments extracted from social networks," *IEEE Trans. Consumer Electronics*, vol. 61, no. 3, pp. 359–367, 2015.

[3] R. Y. K. Lau, Y. Xia, and Y. Ye, "A probabilistic generative model for mining cybercriminal networks from online social media," *IEEE Computational Intelligence Magazine*, vol. 9, no. 1, pp. 31–43, 2014.

[4] X. Li, H. Xie, L. Chen, J. Wang, and X. Deng, "News impact on stock price return via sentiment analysis," *Knowledge-Based Systems*, vol. 69, pp. 14–23, 2014.

[5] S. Rill, D. Reinel, J. Scheidt, and R. V. Zicari, "Politwi: Early detection of emerging political topics on twitter and the impact on concept-level sentiment analysis," *Knowledge-Based Systems*, vol. 69, pp. 24–33, 2014.

[6] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2002, pp. 417–424.

[7] B. Liu, *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers, 2012, vol. 5, no. 1.

[8] S. Poria, E. Cambria, G. Winterstein, and G.-B. Huang, "Sentic patterns: Dependency-based rules for concept-level sentiment analysis," *Knowledge-Based Systems*, vol. 69, pp. 45–63, 2014.

[9] E. Cambria and B. White, "Jumping NLP curves: A review of natural language processing research," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, 2014.

[10] W. Gao, N. Kaji, N. Yoshinaga, and M. Kitsuregawa, "Collective sentiment classification based on user leniency and product popularity," *Journal of Natural Language Processing*, vol. 21, no. 3, pp. 541–561, 2014.

[11] D. Tang, B. Qin, and T. Liu, "Learning semantic representations of users and products for document level sentiment classification," in *Proceedings of the 53nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[12] F. Li, N. Liu, H. Jin, K. Zhao, Q. Yang, and X. Zhu, "Incorporating reviewer and product information for review rating prediction," in *Proceedings of the 22th international joint conference on Artical intelligence (IJCAI)*, vol. 11, 2011, pp. 1820–1825.

[13] Y. Seroussi, I. Zukerman, and F. Bohnert, "Collaborative inference of sentiments from texts," in *User Modeling, Adaptation, and Personalization*. Springer, 2010, pp. 195–206.

[14] Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS)," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 193–202.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of Workshop at the International Conference on Learning Representations (ICLR)*, 2013.

[16] Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” in *Proceedings of the 31th International Conference on Machine Learning (ICML)*, 2014, pp. 1188–1196.

[17] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013, pp. 1631–1642.

[18] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, “A convolutional neural network for modelling sentences,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 655–665.

[19] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, October 2014, pp. 1746–1751.

[20] D. Tang, B. Qin, T. Liu, and Y. Yang, “User modeling with neural network for review rating prediction,” in *Proceedings of the 24th international joint conference on Artical intelligence (IJCAI)*, 2015, pp. 1340–1346.

[21] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[22] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[23] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.

[24] T. Mullen and N. Collier, “Sentiment analysis using support vector machines with diverse information sources,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, vol. 4, 2004, pp. 412–418.

[25] A. B. Goldberg and X. Zhu, “Seeing stars when there aren’t many stars: graph-based semi-supervised learning for sentiment categorization,” in *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 45–52.

[26] T. Nakagawa, K. Inui, and S. Kurohashi, “Dependency tree-based sentiment classification using CRFs with hidden variables,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics, 2010, pp. 786–794.

[27] A. Bickerstaffe and I. Zukerman, “A hierarchical classifier applied to multi-way sentiment detection,” in *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 62–70.

[28] S. Kiritchenko, X. Zhu, and S. M. Mohammad, “Sentiment analysis of short informal texts,” *Journal of Artificial Intelligence Research*, vol. 50, no. 1, pp. 723–762, 2014.

[29] A. Gangemi, V. Presutti, and D. Reforgiato Recupero, “Frame-based detection of opinion holders and topics: a model and a tool,” *Computational Intelligence Magazine, IEEE*, vol. 9, no. 1, pp. 20–30, 2014.

[30] R. Xu, L. Gui, J. Xu, Q. Lu, and K.-F. Wong, “Cross lingual opinion holder extraction based on multi-kernel SVMs and transfer learning,” *World Wide Web*, vol. 18, no. 2, pp. 299–316, 2015.

[31] E. Cambria, “Affective computing and sentiment analysis,” *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 102–107, 2016.

[32] S. Poria, E. Cambria, A. Gelbukh, F. Bisio, and A. Hussain, “Sentiment data flow analysis by means of dynamic linguistic patterns,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 26–36, 2015.

[33] C. Strapparava, A. Valitutti *et al.*, “WordNet Affect: an affective extension of WordNet.” in *Proceedings of the 4th Intl Conf. Language Resources and Evaluation*, vol. 4, 2004, pp. 1083–1086.

[34] A. Esuli and F. Sebastiani, “Sentiwordnet: A publicly available lexical resource for opinion mining,” in *Proceedings of the 5th Intl Conf. Language Resources and Evaluation*, vol. 6. Citeseer, 2006, pp. 417–422.

[35] E. Cambria and A. Hussain, *Sentic computing: a common-sense-based framework for concept-level sentiment analysis*. Cham, Switzerland: Springer, 2015, vol. 1.

[36] G. E. Hinton, “Learning distributed representations of concepts,” in *Proceedings of CogSci*, vol. 1. Amherst, MA, 1986, pp. 1–12.

[37] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, “Improving word representations via global context and multiple word prototypes,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*. Association for Computational Linguistics, 2012, pp. 873–882.

[38] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, “Efficient non-parametric estimation of multiple embeddings per word in vector space,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1059–1069.

[39] F. Tian, H. Dai, J. Bian, B. Gao, R. Zhang, E. Chen, and T.-Y. Liu, “A probabilistic model for learning multi-prototype word embeddings,” in *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, 2014, pp. 151–160.

[40] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 9, pp. 533–536, 1986.

[41] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[42] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proceedings of the 25th International Conference on Machine Learning (ICML)*. ACM, 2008, pp. 160–167.

[43] R. Socher, C. D. Manning, and A. Y. Ng, “Learning continuous phrase representations and syntactic parsing with recursive neural networks,” in *Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010, pp. 1–9.

[44] J. Zhang, S. Liu, M. Li, M. Zhou, and C. Zong, “Bilingually-constrained phrase embeddings for machine translation,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 111–121.

[45] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, “Recurrent neural network based language model.” in *Proceedings of INTERSPEECH*, 2010, pp. 1045–1048.

[46] Y. Ji and J. Eisenstein, “Representation learning for text-level discourse parsing,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 13–24.

[47] A. L. Maas and A. Y. Ng, “A probabilistic model for semantic word vectors,” in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[48] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, “Sentiment embeddings with applications to sentiment analysis,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 28, no. 2, pp. 496–509, 2015.

[49] T. Chen, R. Xu, Y. He, and X. Wang, “A gloss composition and context clustering based distributed word sense representation model,” *Entropy*, vol. 17, no. 9, pp. 6007–6024, 2015.

[50] G. A. Miller, “Wordnet: a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[51] D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1422–1432.

[52] W. Gao, N. Yoshinaga, N. Kaji, and M. Kitsuregawa, “Modeling user leniency and product popularity for sentiment classification,” *Proceedings of IJCNLP, Nagoya, Japan. to appear*, 2013.

[53] F. Li, S. Wang, S. Liu, and M. Zhang, “Suit: A supervised user-item based topic model for sentiment analysis,” in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 1636–1642.

[54] Y. Zhang, H. Zhang, M. Zhang, Y. Liu, and S. Ma, “Do users rate or review?: boost phrase-level sentiment labeling with review-level sentiment classification,” in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 1027–1030.

[55] Y. Mao and G. Lebanon, “Sequential models for sentiment prediction,” in *ICML Workshop on Learning in Structured Output Spaces*, 2006.

[56] X. Liu and M. Zhou, "Sentence-level sentiment analysis via sequence modeling," in *Applied Informatics and Communication*. Springer, 2011, pp. 337–343.

[57] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.

[58] R. Johansson and A. Moschitti, "Syntactic and semantic structure for opinion expression detection," in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2010, pp. 67–76.

[59] B. Yang and C. Cardie, "Extracting opinion expressions with semi-markov conditional random fields," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 1335–1345.

[60] O. Irsoy and C. Cardie, "Opinion mining with deep recurrent neural networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, October 2014, pp. 720–728.

[61] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[62] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[63] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[64] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[65] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[66] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," *Diploma, Technische Universität München*, 1991.

[67] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[68] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 1555–1565.

[69] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 20, no. 2, pp. 215–242, 1958.

[70] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.