Document Modeling with Gated Recurrent Neural Network for Sentiment Classification

Duyu Tang, Bing Qin*, Ting Liu
Harbin Institute of Technology, Harbin, China
{dytang, qinb, tliu}@ir.hit.edu.cn

Abstract

Document level sentiment classification remains a challenge: encoding the intrinsic relations between sentences in the semantic meaning of a document. To address this, we introduce a neural network model to learn vector-based document representation in a unified, bottom-up fashion. The model first learns sentence representation with convolutional neural network or long short-term memory. Afterwards, semantics of sentences and their relations are adaptively encoded in document representation with gated recurrent neural network. We conduct document level sentiment classification on four large-scale review datasets from IMDB and Yelp Dataset Challenge. Experimental results show that: (1) our neural model shows superior performances over several state-of-the-art algorithms; (2) gated recurrent neural network dramatically outperforms standard recurrent neural network in document modeling for sentiment classification.[1]

1 Introduction

Document level sentiment classification is a fundamental task in sentiment analysis, and is crucial to understand user generated content in social networks or product reviews (Manning and Schütze, 1999; Jurafsky and Martin, 2000; Pang and Lee, 2008; Liu, 2012). The task calls for identifying the overall sentiment polarity (e.g. *thumbs up* or *thumbs down*, 1-5 stars on review sites) of a document. In literature, dominant approaches follow (Pang et al., 2002) and exploit machine learning algorithm to build sentiment classifier. Many of them focus on designing hand-crafted features (Qu et al., 2010; Paltoglou and Thelwall, 2010) or learning discriminate features from data, since the performance of a machine learner is heavily dependent on the choice of data representation (Bengio et al., 2015).

Document level sentiment classification remains a significant challenge: how to encode the intrinsic (semantic or syntactic) relations between sentences in the semantic meaning of document. This is crucial for sentiment classification because relations like "contrast" and "cause" have great influences on determining the meaning and the overall polarity of a document. However, existing studies typically fail to effectively capture such information. For example, Pang et al. (2002) and Wang and Manning (2012) represent documents with bag-of-ngrams features and build SVM classifier upon that. Although such feature-driven SVM is an extremely strong performer and hardly to be transcended, its "sparse" and "discrete" characteristics make it clumsy in taking into account of side information like relations between sentences. Recently, Le and Mikolov (2014) exploit neural networks to learn continuous document representation from data. Essentially, they use local ngram information and do not capture semantic relations between sentences. Furthermore, a person asked to do this task will naturally carry it out in a sequential, bottom-up fashion, analyze the meanings of sentences before considering semantic relations between them. This motivates us to develop an end-to-end and bottom-up algorithm to effectively model document representation.

In this paper, we introduce a neural network approach to learn continuous document representation for sentiment classification. The method is on the basis of the principle of compositionality (Frege, 1892), which states that the meaning of a longer expression (e.g. a sentence or a docu-

*Corresponding author.

[1] Codes and datasets are publicly available at http://ir.hit.edu.cn/~dytang.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1422–1432,
Lisbon, Portugal, 17-21 September 2015.

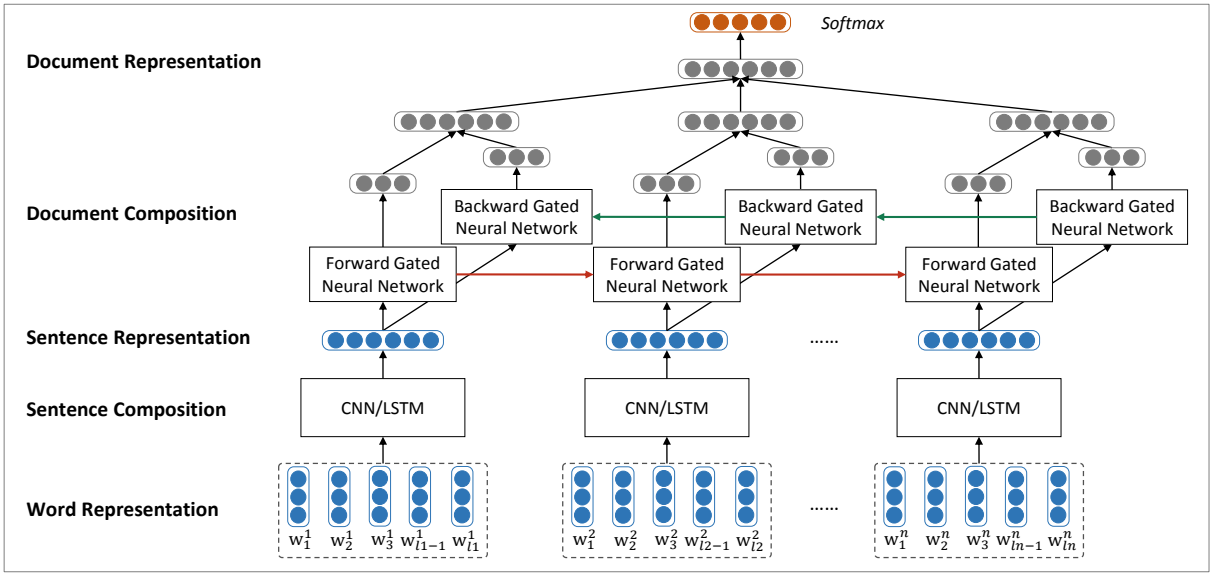

Figure 1: The neural network model for document level sentiment classification. w_i^n stands for the i-th word in the n-th sentence, l_n is sentence length.

ment) depends on the meanings of its constituents. Specifically, the approach models document representation in two steps. In the first step, it uses convolutional neural network (CNN) or long short-term memory (LSTM) to produce sentence representations from word representations. Afterwards, gated recurrent neural network is exploited to adaptively encode semantics of sentences and their inherent relations in document representations. These representations are naturally used as features to classify the sentiment label of each document. The entire model is trained end-to-end with stochastic gradient descent, where the loss function is the cross-entropy error of supervised sentiment classification[2].

We conduct document level sentiment classification on four large-scale review datasets from IMDB[3] and Yelp Dataset Challenge[4]. We compare to neural network models such as paragraph vector (Le and Mikolov, 2014), convolutional neural network, and baselines such as feature-based SVM (Pang et al., 2002), recommendation algorithm JMARS (Diao et al., 2014). Experimental results show that: (1) the proposed neural model shows superior performances over all baseline algorithms; (2) gated recurrent neural network dramatically outperforms standard recurrent neural network in document modeling. The main contributions of this work are as follows:

- We present a neural network approach to encode relations between sentences in document representation for sentiment classification.
- We report empirical results on four large-scale datasets, and show that the approach outperforms state-of-the-art methods for document level sentiment classification.
- We report empirical results that traditional recurrent neural network is weak in modeling document composition, while adding neural gates dramatically improves the classification performance.

2 The Approach

We introduce the proposed neural model in this section, which computes continuous vector representations for documents of variable length. These representations are further used as features to classify the sentiment label of each document. An overview of the approach is displayed in Figure 1.

Our approach models document semantics based on the principle of compositionality (Frege, 1892), which states that the meaning of a longer expression (e.g. a sentence or a document) comes from the meanings of its constituents and the rules used to combine them. Since a document consists of a list of sentences and each sentence is made up of a list of words, the approach models document representation in two stages. It first produces continuous sentence vectors from word represen-

[2]A similar work can be found at: http://deeplearning.net/tutorial/lstm.html

[3]http://www.imdb.com/

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

tations with sentence composition (Section 2.1). Afterwards, sentence vectors are treated as inputs of document composition to get document representation (Section 2.2). Document representations are then used as features for document level sentiment classification (Section 2.3).

2.1 Sentence Composition

We first describe word vector representation, before presenting a convolutional neural network with multiple filters for sentence composition.

Each word is represented as a low dimensional, continuous and real-valued vector, also known as word embedding (Bengio et al., 2003). All the word vectors are stacked in a word embedding matrix $L_w \in \mathbb{R}^{d \times |V|}$, where d is the dimension of word vector and $|V|$ is vocabulary size. These word vectors can be randomly initialized from a uniform distribution (Socher et al., 2013b), or be pre-trained from text corpus with embedding learning algorithms (Mikolov et al., 2013; Pennington et al., 2014; Tang et al., 2014). We adopt the latter strategy to make better use of semantic and grammatical associations of words.

We use convolutional neural network (CNN) and long short-term memory (LSTM) to compute continuous representations of sentences with semantic composition. CNN and LSTM are state-of-the-art semantic composition models for sentiment classification (Kim, 2014; Kalchbrenner et al., 2014; Johnson and Zhang, 2015; Li et al., 2015a). They learn fixed-length vectors for sentences of varying length, captures words order in a sentence and does not depend on external dependency or constituency parse results. One could also use tree-based composition method such as Recursive Neural Tensor Network (Socher et al., 2013b) or Tree-Structured LSTM (Tai et al., 2015; Zhu et al., 2015) as alternatives.

Specifically, we try CNN with multiple convolutional filters of different widths (Tang et al., 2015) to produce sentence representation. Figure 2 displays the method. We use multiple convolutional filters in order to capture local semantics of n-grams of various granularities, which have been proven effective for sentiment classification. For example, a convolutional filter with a width of 2 essentially captures the semantics of bigrams in a sentence. In this work, we use three convolutional filters whose widths are 1, 2 and 3 to encode the semantics of unigrams, bigrams and trigrams in a sentence. Each filter consists of a list of linear layers with shared parameters. Formally, let us denote a sentence consisting of n words as $\{w_1, w_2, ...w_i, ...w_n\}$, let l_c be the width of a convolutional filter, and let W_c, b_c be the shared parameters of linear layers in the filter. Each word w_i is mapped to its embedding representation $e_i \in \mathbb{R}^d$. The input of a linear layer is the concatenation of word embeddings in a fixed-length window size l_c, which is denoted as $I_c = [e_i; e_{i+1}; ...; e_{i+l_c-1}] \in \mathbb{R}^{d \cdot l_c}$. The output of a linear layer is calculated as

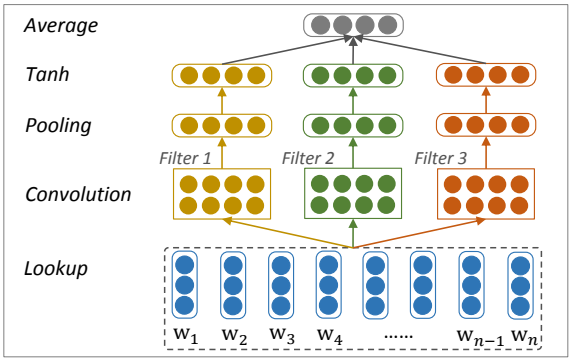

Figure 2: Sentence composition with convolutional neural network.

$$O_c = W_c \cdot I_c + b_c \tag{1}$$

where $W_c \in \mathbb{R}^{l_{oc} \times d \cdot l_c}$, $b_c \in \mathbb{R}^{l_{oc}}$, l_{oc} is the output length of linear layer. To capture global semantics of a sentence, we feed the outputs of linear layers to an *average* pooling layer, resulting in an output vector with fixed-length. We further add hyperbolic tangent ($tanh$) to incorporate pointwise nonlinearity, and *average* the outputs of multiple filters to get sentence representation.

We also try lstm as the sentence level semantic calculator, the performance comparison between these two variations is given in Section 3.

2.2 Document Composition with Gated Recurrent Neural Network

The obtained sentence vectors are fed to a document composition component to calculate the document representation. We present a gated recurrent neural network approach for document composition in this part.

Given the vectors of sentences of variable length as input, document composition produces a fixed-length document vector as output. To this end, a simple strategy is ignoring the order of sen-

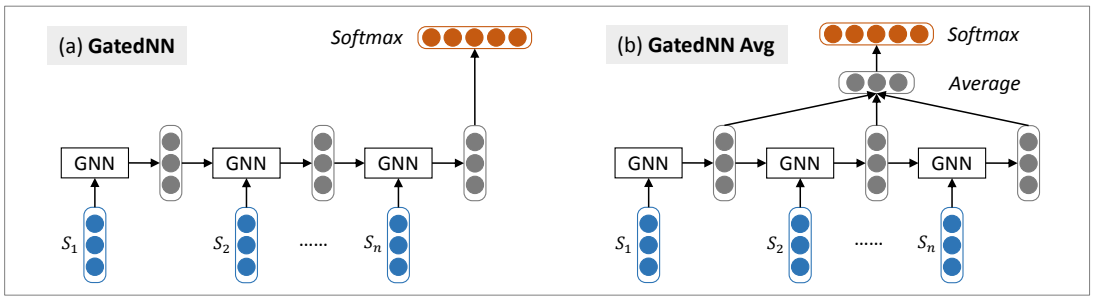

Figure 3: Document modeling with gated recurrent neural network. GNN stands for the basic computational unit of gated recurrent neural network.

tences and averaging sentence vectors as document vector. Despite its computational efficiency, it fails to capture complex linguistic relations (e.g. "cause" and "contrast") between sentences. Convolutional neural network (Denil et al., 2014) is an alternative for document composition, which models local sentence relations with shared parameters of linear layers.

Standard recurrent neural network (RNN) can map vectors of sentences of variable length to a fixed-length vector by recursively transforming current sentence vector s_t with the output vector of the previous step h_{t-1}. The transition function is typically a linear layer followed by pointwise non-linearity layer such as $tanh$.

$$h_t = tanh(W_r \cdot [h_{t-1}; s_t] + b_r) \quad (2)$$

where $W_r \in \mathbb{R}^{l_h \times (l_h + l_{oc})}$, $b_r \in \mathbb{R}^{l_h}$, l_h and l_{oc} are dimensions of hidden vector and sentence vector, respectively. Unfortunately, standard RNN suffers the problem of gradient vanishing or exploding (Bengio et al., 1994; Hochreiter and Schmidhuber, 1997), where gradients may grow or decay exponentially over long sequences. This makes it difficult to model long-distance correlations in a sequence. To address this problem, we develop a gated recurrent neural network for document composition, which works in a sequential way and adaptively encodes sentence semantics in document representations. The approach is analogous to the recently emerged LSTM (Graves et al., 2013; Zaremba and Sutskever, 2014; Sutskever et al., 2014; Xu et al., 2015) and gated neural network (Cho et al., 2014; Chung et al., 2015). Specifically, the transition function of the gated RNN used in this work is calculated as follows.

$$i_t = sigmoid(W_i \cdot [h_{t-1}; s_t] + b_i) \quad (3)$$

$$f_t = sigmoid(W_f \cdot [h_{t-1}; s_t] + b_f) \quad (4)$$

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

$$h_t = tanh(i_t \odot g_t + f_t \odot h_{t-1}) \quad (6)$$

where $\odot$ stands for element-wise multiplication, W_i, W_f, b_i, b_f adaptively select and remove history vector and input vector for semantic composition. The model can be viewed as a LSTM whose output gate is alway on, since we prefer not to discarding any part of the semantics of sentences to get a better document representation. Figure 3 (a) displays a standard sequential way where the last hidden vector is regarded as the document representation for sentiment classification. We can make further extensions such as averaging hidden vectors as document representation, which takes considerations of a hierarchy of historical semantics with different granularities. The method is illustrated in Figure 3 (b), which shares some characteristics with (Zhao et al., 2015). We can go one step further to use preceding histories and following evidences in the same way, and exploit bidirectional (Graves et al., 2013) gated RNN as the calculator. The model is embedded in Figure 1.

2.3 Sentiment Classification

The composed document representations can be naturally regarded as features of documents for sentiment classification without feature engineering. Specifically, we first add a linear layer to transform document vector to real-valued vector whose length is class number C. Afterwards, we add a $softmax$ layer to convert real values to conditional probabilities, which is calculated as follows.

$$P_i = \frac{exp(x_i)}{\sum_{i'=1}^{C} exp(x_{i'})} \quad (7)$$

We conduct experiments in a supervised learning setting, where each document in the training data is accompanied with its gold sentiment label.

Corpus	#docs	#s/d	#w/d	$\|V\|$	#class	Class Distribution
Yelp 2013	335,018	8.90	151.6	211,245	5	.09/.09/.14/.33/.36
Yelp 2014	1,125,457	9.22	156.9	476,191	5	.10/.09/.15/.30/.36
Yelp 2015	1,569,264	8.97	151.9	612,636	5	.10/.09/.14/.30/.37
IMDB	348,415	14.02	325.6	115,831	10	.07/.04/.05/.05/.08/.11/.15/.17/.12/.18

Table 1: Statistical information of Yelp 2013/2014/2015 and IMDB datasets. #docs is the number of documents, #s/d and #w/d represent average number of sentences and average number of words contained in per document, $|V|$ is the vocabulary size of words, #class is the number of classes.

For model training, we use the cross-entropy error between gold sentiment distribution $P^g(d)$ and predicted sentiment distribution $P(d)$ as the loss function.

$$loss = -\sum_{d \in T} \sum_{i=1}^{C} P_i^g(d) \cdot log(P_i(d)) \quad (8)$$

where T is the training data, C is the number of classes, d represents a document. $P^g(d)$ has a *1-of-K* coding scheme, which has the same dimension as the number of classes, and only the dimension corresponding to the ground truth is 1, with all others being 0. We take the derivative of loss function through back-propagation with respect to the whole set of parameters $\theta = [W_c; b_c; W_i; b_i; W_f; b_f; W_r; b_r; W_{softmax}, b_{softmax}]$, and update parameters with stochastic gradient descent. We set the widths of three convolutional filters as 1, 2 and 3, output length of convolutional filter as 50. We learn 200-dimensional word embeddings with SkipGram (Mikolov et al., 2013) on each dataset separately, randomly initialize other parameters from a uniform distribution $U(-0.01, 0.01)$, and set learning rate as 0.03.

3 Experiment

We conduct experiments to empirically evaluate our method by applying it to document level sentiment classification. We describe experimental settings and report empirical results in this section.

3.1 Experimental Setting

We conduct experiments on large-scale datasets consisting of document reviews. Specifically, we use one movie review dataset from IMDB (Diao et al., 2014) and three restaurant review datasets from Yelp Dataset Challenge in 2013, 2014 and 2015. Human labeled review ratings are regarded as gold standard sentiment labels, so that we do not need to manually annotate sentiment labels of documents. We do not consider the cases that rating does not match with review texts (Zhang et al., 2014).

Statistical information of these datasets are given in Table 1. We use the same dataset split as in (Diao et al., 2014) on IMDB dataset, and split Yelp datasets into training, development and testing sets with 80/10/10. We run tokenization and sentence splitting with Stanford CoreNLP (Manning et al., 2014) on all these datasets. We use *accuracy* (Manning and Schütze, 1999; Jurafsky and Martin, 2000) and MSE (Diao et al., 2014) as evaluation metrics, where *accuracy* is a standard metric to measure the overall sentiment classification performance. We use *MSE* to measure the divergences between predicted sentiment labels and ground truth sentiment labels because review labels reflect sentiment strengths (e.g. one star means strong negative and five star means strong positive).

$$MSE = \frac{\sum_i^N (gold_i - predicted_i)^2}{N} \quad (9)$$

3.2 Baseline Methods

We compare our methods (Conv-GRNN and LSTM-GRNN) with the following baseline methods for document level sentiment classification.

(1) *Majority* is a heuristic baseline, which assigns the majority sentiment label in training set to each document in test set.

(2) In *SVM+Ngrams*, we use bag-of-unigrams and bag-of-bigrams as features and train SVM classifier with LibLinear (Fan et al., 2008)[5].

(3) In *TextFeatures*, we implement sophisticated features (Kiritchenko et al., 2014) including word ngrams, character ngrams, sentiment lexicon features, cluster features, et al.

[5]We also try discretized regression (Pang and Lee, 2005) with fixed decision thresholds (e.g. 0.5, 1.5, 2.5, ...). However, its performance is obviously worse than SVM classifier.

	Yelp 2013		Yelp 2014		Yelp 2015		IMDB	
	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE
Majority	0.356	3.06	0.361	3.28	0.369	3.30	0.179	17.46
SVM + Unigrams	0.589	0.79	0.600	0.78	0.611	0.75	0.399	4.23
SVM + Bigrams	0.576	0.75	0.616	0.65	0.624	0.63	0.409	3.74
SVM + TextFeatures	0.598	0.68	0.618	0.63	0.624	0.60	0.405	3.56
SVM + AverageSG	0.543	1.11	0.557	1.08	0.568	1.04	0.319	5.57
SVM + SSWE	0.535	1.12	0.543	1.13	0.554	1.11	0.262	9.16
JMARS	N/A	–	N/A	–	N/A	–	N/A	4.97
Paragraph Vector	0.577	0.86	0.592	0.70	0.605	0.61	0.341	4.69
Convolutional NN	0.597	0.76	0.610	0.68	0.615	0.68	0.376	3.30
Conv-GRNN	0.637	0.56	0.655	0.51	0.660	0.50	0.425	**2.71**
LSTM-GRNN	**0.651**	**0.50**	**0.671**	**0.48**	**0.676**	**0.49**	**0.453**	3.00

Table 2: Sentiment classification on Yelp 2013/2014/2015 and IMDB datasets. Evaluation metrics are accuracy (higher is better) and MSE (lower is better). The best method in each setting is in **bold**.

(4) In *AverageSG*, we learn 200-dimensional word vectors with *word2vec*[6] (Mikolov et al., 2013), average word embeddings to get document representation, and train a SVM classifier.

(5) We learn sentiment-specific word embeddings (SSWE), and use max/min/average pooling (Tang et al., 2014) to get document representation.

(6) We compare with a state-of-the-art recommendation algorithm JMARS (Diao et al., 2014), which utilizes user and aspects of a review with collaborative filtering and topic modeling.

(7) We implement a convolutional neural network (CNN) baseline as it is a state-of-the-art semantic composition method for sentiment analysis (Kim, 2014; Denil et al., 2014).

(8) We implement a state-of-the-art neural network baseline Paragraph Vector (Le and Mikolov, 2014) because its codes are not officially provided. Window size is tuned on the development set.

3.3 Comparison to Other Methods

Experimental results are given in Table 2. We evaluate each dataset with two metrics, namely accuracy (higher is better) and MSE (lower is better). The best method in each dataset and each evaluation metric is in bold.

From Table 2, we can see that majority is the worst method because it does not capture any textual semantics. SVM classifiers with unigram and bigram features (Pang et al., 2002) are extremely strong, which are almost the strongest performers among all baseline methods. Designing complex features are also effective for document level sentiment classification, however, it does not surpass the bag-of-ngram features significantly as on Twitter corpora (Kiritchenko et al., 2014). Furthermore, the aforementioned bag-of-features are discrete and sparse. For example, the feature dimension of bigrams and TextFeatures on Yelp 2015 dataset are 899K and 4.81M after we filter out low frequent features. Based on them, we try to concatenate several discourse-driven features, but the classification performances remain unchanged.

AverageSG is a straight forward way to compose document representation without feature engineering. Unfortunately, we can see that it does not work in this scenario, which appeals for powerful semantic composition models for document level sentiment classification. We try to make better use of the sentiment information to learn a better SSWE (Tang et al., 2014), e.g. setting a large window size. However, its performance is still worse than context-based word embedding. This stems from the fact that there are many sentiment shifters (e.g. negation or contrast words) in document level reviews, while Tang et al. (2014) learn SSWE by assigning sentiment label of a text to each phrase it contains. How to learn SSWE effectively with document level sentiment supervision remains as an interesting future work.

Since JMARS outputs real-valued outputs, we only evaluate it in terms of MSE. We can see that sophisticated baseline methods such as JMARS, paragraph vector and convolutional NN obtain significant performance boosts over AverageSG by

[6]We use Skipgram as it performs slightly better than CBOW in the experiment. We also try off-the-shell word embeddings from Glove, but its performance is slightly worse than tailored word embedding from each corpus.

	Yelp 2013		Yelp 2014		Yelp 2015		IMDB	
	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE	Accuracy	MSE
Average	0.598	0.65	0.605	0.75	0.614	0.67	0.366	3.91
Recurrent	0.377	1.37	0.306	1.75	0.383	1.67	0.176	12.29
Recurrent Avg	0.582	0.69	0.591	0.70	0.597	0.74	0.344	3.71
Bi Recurrent Avg	0.587	0.73	0.597	0.73	0.577	0.82	0.372	3.32
GatedNN	0.636	0.58	0.656	0.52	0.651	0.51	**0.430**	2.95
GatedNN Avg	0.635	0.57	**0.659**	0.52	0.657	0.56	0.416	2.78
Bi GatedNN Avg	**0.637**	**0.56**	0.655	**0.51**	**0.660**	**0.50**	0.425	**2.71**

Table 3: Sentiment classification on IMDB, Yelp 2013/2014/2015 datasets. Evaluation metrics are accuracy (higher is better) and MSE (lower is better). The best method in each setting is in **bold**.

capturing deeper semantics of texts. Comparing between CNN and AverageSG, we can conclude that deep semantic compositionality is crucial for understanding the semantics and the sentiment of documents. However, it is somewhat disappointing that these models do not significantly outperform discrete bag-of-ngrams and bag-of-features. The reason might lie in that semantic meanings of documents, e.g. relations between sentences, are not well captured. We can see that the proposed method Conv-GRNN and LSTM-GRNN yield the best performance on all four datasets in two evaluation metrics. Compared with CNN, Conv-GRNN shows its superior power in document composition component, which encodes semantics of sentences and their relations in document representation with gated recurrent neural network. We also find that LSTM (almost) consistently performs better than CNN in modeling the sentence representation.

3.4 Model Analysis

As discussed before, document composition contributes a lot to the superior performance of Conv-GRNN and LSTM-GRNN. Therefore, we take Conv-GRNN as an example and compare different neural models for document composition in this part. Specifically, after obtaining sentence vectors with convolutional neural network as described in Section 2.1, we carry out experiments in following settings.

(1) *Average*. Sentence vectors are averaged to get the document vector.

(2) *Recurrent* / *GatedNN*. Sentence vectors are fed to standard (or gated) recurrent neural network in a sequential way from the beginning of the input document. The last hidden vector is regarded as document representation.

(3) *Recurrent Avg* / *GatedNN Avg*. We extend setting (2) by averaging hidden vectors of recurrent neural network as document vector.

(4) *Bi Recurrent Avg* / *Bi GatedNN Avg*. We extend setting (3) by calculating hidden vectors from both preceding histories and following contexts. Bi-directional hidden vectors are averaged as document representation.

Table 3 shows the experimental results. We can see that standard recurrent neural network (RNN) is the worst method, even worse than the simple vector average. This is because RNN suffers from the vanishing gradient problem, stating that the influence of a given input on the hidden layer decays exponentially over time on the network output. In this paper, it means that document representation encodes rare semantics of the beginning sentences. This is further justified by the great improvement of *Recurrent Avg* over *Recurrent*. *Bi Recurrent Avg* and *Recurrent Avg* perform comparably, but disappointingly both of them fail to transcend *Average*. After adding neural gates, *GatedNN* obtains dramatic accuracy improvements over *Recurrent* and significantly outperforms previous settings. The results indicate that *Gated RNN* is capable of handling the vanishing gradient problem to some extend, and it is practical to adaptively model sentence semantics in document representation. *GatedNN Avg* and *Bi GatedNN Avg* obtains comparable performances with *GatedNN*.

4 Related Work

Document level sentiment classification is a fundamental problem in sentiment analysis (Pang and Lee, 2008; Liu, 2012), which aims at identifying the sentiment label of a document (Pang et al., 2002; Turney, 2002). Pang and Lee (2002; 2005)

cast this problem as a classification task, and use machine learning method in a supervised learning framework. Turney (2002) introduces an unsupervised approach by using sentiment words/phrases extracted from syntactic patterns to determine the document polarity. Goldberg and Zhu (2006) place this task in a semi-supervised setting, and use unlabelled reviews with graph-based method. Dominant studies in literature follow Pang et al. (2002) and work on designing effective features for building a powerful sentiment classifier. Representative features include word ngrams (Wang and Manning, 2012), text topic (Ganu et al., 2009), bag-of-opinions (Qu et al., 2010), syntactic relations (Xia and Zong, 2010), sentiment lexicon features (Kiritchenko et al., 2014).

Despite the effectiveness of feature engineering, it is labor intensive and unable to extract and organize the discriminative information from data (Bengio et al., 2015). Recently, neural network emerges as an effective way to learn continuous text representation for sentiment classification. Existing studies in this direction can be divided into two groups. One line of research focuses on learning continuous word embedding. Traditional embedding learning algorithms typically leverage contexts of words in a context-prediction way (Bengio et al., 2003; Mikolov et al., 2013; Baroni et al., 2014). Since these methods typically map words with similar contexts but opposite polarity (e.g. "good" and "bad") to neighboring vectors, several studies (Maas et al., 2011; Labutov and Lipson, 2013; Tang et al., 2014) learn sentiment-specific word embeddings by taking sentiment of texts into account. Another line of research concentrates on semantic composition (Mitchell and Lapata, 2010). Yessenalina and Cardie (2011) represent each word as a matrix and use iterated matrix multiplication as phrase-level composition function. Socher et al. (2013b) introduce a family of recursive neural networks for sentence-level semantic composition. Recursive neural network is extended with global feedbackward (Paulus et al., 2014), feature weight tuning (Li, 2014), deep recursive layer (Irsoy and Cardie, 2014), adaptive composition functions (Dong et al., 2014), combined with Combinatory Categorial Grammar (Hermann and Blunsom, 2013), and used for opinion relation detection (Xu et al., 2014). Glorot et al. (2011) use stacked denoising autoencoder. Convolutional neural networks are widely used for semantic composition (Kim, 2014; Kalchbrenner et al., 2014; Denil et al., 2014; Johnson and Zhang, 2015) by automatically capturing local and global semantics. Le and Mikolov (2014) introduce Paragraph Vector to learn document representation from semantics of words. Sequential model like recurrent neural network or long short-term memory (LSTM) are also verified as strong approaches for semantic composition (Li et al., 2015a).

In this work, we represent document with convolutional-gated recurrent neural network, which adaptively encodes semantics of sentences and their relations. A recent work in (Li et al., 2015b) also investigate LSTM to model document meaning. They verify the effectiveness of LSTM in text generation task.

5 Conclusion

We introduce neural network models (Conv-GRNN and LSTM-GRNN) for document level sentiment classification. The approach encodes semantics of sentences and their relations in document representation, and is effectively trained end-to-end with supervised sentiment classification objectives. We conduct extensive experiments on four review datasets with two evaluation metrics. Empirical results show that our approaches achieve state-of-the-art performances on all these datasets. We also find that (1) traditional recurrent neural network is extremely weak in modeling document composition, while adding neural gates dramatically boosts the performance, (2) LSTM performs better than a multi-filtered CNN in modeling sentence representation.

We briefly discuss some future plans. How to effectively compose sentence meanings to document meaning is a central problem in natural language processing. In this work, we develop neural models in a sequential way, and encode sentence semantics and their relations automatically without using external discourse analysis results. From one perspective, one could carefully define a set of sentiment-sensitive discourse relations (Zhou et al., 2011), such as "contrast", "condition", "cause", etc. Afterwards, relation-specific gated RNN can be developed to explicitly model semantic composition rules for each relation (Socher et al., 2013a). However, defining such a relation scheme is linguistic driven and time consuming, which we leave as future work. From another perspective, one could compose document

representation over discourse tree structures rather than in a sequential way. Accordingly, Recursive Neural Network (Socher et al., 2013b) and Structured LSTM (Tai et al., 2015; Zhu et al., 2015) can be used as composition algorithms. However, existing discourse structure learning algorithms are difficult to scale to massive review texts on the web. How to simultaneously learn document structure and composition function is an interesting future work.

Acknowledgments

The authors give great thanks to Yaming Sun and Jiwei Li for the fruitful discussions. We also would like to thank three anonymous reviewers for their valuable comments and suggestions. This work was supported by the National High Technology Development 863 Program of China (No. 2015AA015407), National Natural Science Foundation of China (No. 61133012 and No. 61273321). Duyu Tang is supported by Baidu Fellowship and IBM Ph.D. Fellowship.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL*, pages 238–247.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Yoshua Bengio, Ian J. Goodfellow, and Aaron Courville. 2015. Deep learning. Book in preparation for MIT Press.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. *ICML*.

Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, and Nando de Freitas. 2014. Modelling, visualising and summarising documents with a single convolutional neural network. *arXiv preprint:1406.3830*.

Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *SIGKDD*, pages 193–202. ACM.

Li Dong, Furu Wei, Ming Zhou, and Ke Xu. 2014. Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis. In *AAAI*, pages 1537–1543.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *JMLR*.

Gottlob Frege. 1892. On sense and reference. *Ludlow (1997)*, pages 563–584.

Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. Beyond the stars: Improving rating predictions using review text content. In *WebDB*.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520.

Andrew B Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *GraphBased Method for NLP*, pages 45–52.

Alex Graves, Navdeep Jaitly, and A-R Mohamed. 2013. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE.

Karl Moritz Hermann and Phil Blunsom. 2013. The role of syntax in vector space models of compositional semantics. In *ACL*, pages 894–904.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In *NIPS*, pages 2096–2104.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. *NAACL*.

Dan Jurafsky and James H Martin. 2000. *Speech & language processing*. Pearson Education India.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *ACL*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, pages 723–762.

Igor Labutov and Hod Lipson. 2013. Re-embedding words. In *Annual Meeting of the Association for Computational Linguistics*.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196.

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

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

Jiwei Li. 2014. Feature weight tuning for recursive neural networks. *Arxiv preprint*, 1412.3714.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT press.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL*, pages 55–60.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429.

Georgios Paltoglou and Mike Thelwall. 2010. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pages 1386–1395.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP*, pages 79–86.

Romain Paulus, Richard Socher, and Christopher D Manning. 2014. Global belief recursive neural networks. In *NIPS*, pages 2888–2896.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In *COLING*, pages 913–921.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013a. Parsing with compositional vector grammars. In *ACL*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL*, pages 1555–1565.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, pages 1014–1023.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, pages 90–94.

Rui Xia and Chengqing Zong. 2010. Exploring the use of word relation features for sentiment classification. In *COLING*, pages 1336–1344.

Liheng Xu, Kang Liu, and Jun Zhao. 2014. Joint opinion relation detection using one-class deep neural network. In *COLING*, pages 677–687.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *ICML*.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *EMNLP*, pages 172–182.

Wojciech Zaremba and Ilya Sutskever. 2014. Learning to execute. *arXiv preprint arXiv:1410.4615*.

Yongfeng Zhang, Haochen Zhang, Min Zhang, Yiqun Liu, and Shaoping Ma. 2014. Do users rate or review?: boost phrase-level sentiment labeling with review-level sentiment classification. In *SIGIR*, pages 1027–1030. ACM.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *IJCAI*.

Lanjun Zhou, Binyang Li, Wei Gao, Zhongyu Wei, and Kam-Fai Wong. 2011. Unsupervised discovery of discourse relations for eliminating intra-sentence polarity ambiguities. In *EMNLP*, pages 162–171, .

Xiaodan Zhu, Parinaz Sobhani, and Hongyu Guo. 2015. Long short-term memory over tree structures. *ICML*.