Applied Soft Computing Journal 94 (2020) 106435

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models

Wei Li [a], Luyao Zhu [a,*], Yong Shi [b], Kun Guo [b], Erik Cambria [a]

[a] School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore
[b] School of Economics and Management, University of Chinese Academy of Sciences, Beijing, 100190, China

ARTICLE INFO

Article history:
Received 11 April 2018
Received in revised form 11 March 2020
Accepted 26 May 2020
Available online 1 June 2020

Keywords:
Sentiment analysis
Long short-term memory
Convolutional neural network
Sentiment padding
Sentiment lexicon

ABSTRACT

Sentiment analysis, which refers to the task of detecting whether a textual item (e.g., a product review and a blog post) expresses a positive or negative opinion in general or about a given entity (e.g., a product, person, or policy), has received increasing attention in recent years. It serves as an important role in natural language processing. User generated content, like tourism reviews, developed dramatically during the past years, generating a large amount of unstructured data from which it is hard to obtain useful information. Due to the changes in textual order, sequence length and complicated logic, it is still a challenging task to predict the exact sentiment polarities of the user reviews, especially for fine-grained sentiment classification. In this paper, we first propose sentiment padding, a novel padding method compared with zero padding, making the input data sample of a consistent size and improving the proportion of sentiment information in each review. Inspired by the most recent studies with respect to neural networks, we propose deep learning based sentiment analysis models named lexicon integrated two-channel CNN–LSTM family models, combining CNN and LSTM/BiLSTM branches in a parallel manner. Experiments on several challenging datasets, like Stanford Sentiment Treebank, demonstrate that the proposed method outperforms many baseline methods.

1. Introduction

Sentiment analysis (SA) has received more attention in recent years. When a person comments on an entity (e.g., a product, person, or policy), automatically predicting sentiment polarities of the textual item (e.g., a product review and a blog post) is known as SA [1]. So far, various methods have been proposed for English and Chinese SA [2–13], making great progress in the text mining research. However, SA still faces problems such as textual order, changeable sequence length, and complicated logic. These problems are more severe in Chinese for lack of segmentation symbols. For instance, blank spaces in English separate sentences into words [14] while there is no such segmentation symbol in Chinese.

SA techniques mainly comprise two types, namely lexicon-based and machine learning approaches [15]. The core idea of a lexicon-based SA method is to build a sentiment lexicon that contains words expressing opinions or sentiments [16]. Then, the proposed sentiment lexicons are further applied to textual item classification. In order to improve the accuracy of sentiment polarity classification, the sentiment lexicons are supposed to cover as many sentiment words as possible. A number of representative SA methods were proposed from the perspective of the sentiment lexicon [17–28]. In a recent study, Wu et al. [29] presented a data-driven sentiment lexicon construction method which got good performance in classifying microblogs. Their method took use of the improved statistical indicator enhanced mutual information (*EMI*) to detect Chinese new words, raising the coverage and quality of sentiment lexicon. Even though sentiment lexicon is no longer fully utilized for SA, it still provides valuable features and is a promising method [30].

Machine learning methods are of vital importance in SA, achieving the state-of-the-art results so far. Advanced neural network models, like [31–34], pursue a high-quality feature engineering and generalization ability through training powerful neural networks. The approaches in [4,12,13,34,35] obtained excellent performances on several benchmark datasets. However, it needs a large number of labeled data to train such models, especially for deep models. Besides, the training process is both time- and memory-consuming. Other machine learning approaches, such as support vector machine [36,37], calculate a hyperplane maximizing the margin between positive and negative samples.

* Correspondence to: 50 Nanyang Ave, 639798, Singapore.
E-mail addresses: wei008@e.ntu.edu.sg (W. Li), luyao001@e.ntu.edu.sg (L. Zhu), yshi@ucas.ac.cn (Y. Shi), guokun@ucas.ac.cn (K. Guo), erik@sentic.net (E. Cambria).

https://doi.org/10.1016/j.asoc.2020.106435
1568-4946/

The long short-term memory (LSTM) model and convolutional neural network (CNN) in [4,34,38,39] have been extensively studied for SA due to their high performances. However, the performances of LSTM and CNN rely on the amount and quality of labeled data. In general, the deeper neural network calls for the larger demand for data. In addition, the quality of word embedding significantly influences the classification results since it measures the relationship among word vectors in vector space.

In 2017, Alain and Bengio [40] put forward some insights regarding the training process of deep models. They proved that the bifurcation toy model and skip connection operation are useful for training deep neural networks. The bifurcation toy model involves a classification model with two parallel branches whose outputs combine later, while skip connection operation refers to adding input information to the intermediate layer of the neural networks. In the operation, the input features fed to intermediate layer are obtained by concatenating the outputs of the previous layer of intermediate layer and input layer. Moreover, Khan et al. [41,42] took sentiment lexicon into consideration and created two lexicon-based machine learning SA approaches. In this case, sentiment lexicons are incorporated into machine learning classifier. Besides, we found that existing research papers combine CNN and LSTM/BiLSTM model in an ordinal manner [43,44]. Motivated by these observations, we design a lexicon integrated two-channel CNN–BiLSTM model, combining CNN and BiLSTM branches in a parallel manner. Domain-specific sentiment lexicon is employed to generate high-quality textual features fed into the CNN–BiLSTM model. BiLSTM and CNN channels are utilized to complement valuable information to each other. LSTM channel is skilled at processing sequential information, while the CNN channel is capable of extracting abstract features from different horizons. Furthermore, skip connection operation makes it easier to train a deep model. Extensive experiments on benchmark datasets demonstrate that the proposed model surpasses many baseline methods. The main contributions of this paper are shown as follows:

1. We first propose sentiment padding, an efficient padding method compared with zero padding, making input data sample of a consistent size and improving the proportion of sentiment information in each review.
2. We further create a lexicon integrated two-channel CNN–BiLSTM model, integrating CNN and BiLSTM in a parallel manner and taking advantage of the sentiment lexicon information.
3. This paper studies the influence of the skip connection operation on the aforementioned two-channel model.
4. Experimental results indicate the superiority of the proposed model on analyzing English movie reviews and Chinese tourism reviews. We also find the performance of two-channel model depends on the interaction between the channels.

The rest of the paper is organized as follows: In Section 2, the related work about deep models, sentiment analysis, and padding operation is introduced. Section 3 elaborates the mechanism of sentiment padding operation and the proposed two-channel model. In Section 4, we conduct sentiment analysis and parameter analysis experiments. Besides, experiment results are discussed and some meaningful conclusions are obtained. We summarize the paper and give some future insights in the last section.

2. Related work

2.1. Sentiment analysis

Existing works on SA are divided into two types. The first type of methods is sentiment lexicon approaches [17–28], and the second type of methods belongs to machine learning approaches [4,12,13,15,34,35].

The first type of methods mainly considers building a high-quality sentiment lexicon which contains as many sentiment words as possible. A high-quality sentiment lexicon ought to cover not only many normal sentiment words [16] but also user-invented sentiment words. For instance, the sentiment lexicon built in [2] contained many user-invented Chinese sentiment words in Microblog. In 2016, Khan et al. [42] built a general purpose sentiment lexicon in a semi-supervised manner. They used Expected Likelihood Estimation Smoothed Odds Ratio to define word semantics, and the well-defined word semantics were integrated with supervised machine learning based model selection approach. Nevertheless, the performances of lexicon-based SA approaches were limited due to the negligence of textual order, sequence length, and complicated logic.

The second type of methods focuses on training a powerful machine learning model to classify textual items. The LSTM model [45,46] and the BiLSTM model [47] take textual order and sequence length into consideration, while the CNN [9] is aimed at exploiting local features within the text. All the models need effective word embedding. However, existing methods cannot completely solve completely the embedding problem for SA. Besides, the design of neural network architecture lacks theoretic guidance. In addition to the deep models, some other methods, such as adapting Naïve Bayes [48] and conditional random field based approaches [49], got good performances on SA.

A part of SA methods attempted to devise effective text embedding to tackle problems like textual order, changeable sequence length and complicated context. In [31], Santos et al. proposed a deep CNN model entitled CharSCNN, where the character-level representation was included for SA on short text. The character-level representation provided complementary information promoting the training of CNN model. Besides the text representation, another portion of researchers pay more attention to powerful architecture of deep models. To this end, a regional CNN–LSTM model [34] was advanced, so that both local information within sentences and long-distance dependency across sentences are considered for classification. The architecture of this model was carefully devised so as to combine the advantages of both CNN and LSTM models. Yoon and Kim [50] raised a SA framework combining CNN and BiLSTM in an ordinal manner. They first generated a multi-channel word embedding from the perspective of word2vec (static & non-static) and sentiment lexicon. Then, a CNN layer was employed to extract local features, and fed these features into BiLSTM layer. Extensive experiments testified the superiority of this model in comparison with several baselines. Similarly, Wang et al. [51] put forward a combination of convolutional and recurrent neural networks for SA, where CNN and LSTM layers were organized in an ordinal manner. Experimental results showed that the model obtained the state-of-the-art result on Stanford Sentiment Treebank dataset. In addition, the architecture of deep models were tuned to deal with SA tasks in specific domains or languages. A specific deep learning model [5] was presented to perform SA on Arabic. Features of this model originated from newly developed Arabic sentiment lexicon and other standard lexicons. Van et al. [52] proposed a novel framework combining convolutional and recursive neural network for sentence-level SA, which was aimed at alleviating weaknesses of these two models. In 2018, Chen et al. [53] proposed a deep learning model utilizing visual and textual information as different channels for SA tasks. They developed two different CNN models to extract visual and textual information from image and text respectively and their model gained good results on several datasets.

In recent years, sentiment lexicon was introduced into machine learning based SA approaches. Khan et al. [41] came up with a hybrid approach named SWIMS incorporating machine learning with a lexicon based approach, where SentiWordNet determined the feature weight and *SVM* was utilized to learn the feature weights. Poria et al. [54] proposed a SA system based on dependency rules and machine learning classifier. The system took semantic parser to parse text into concepts and then embedded the concepts into vector space. If the concepts were found in SenticNet, sentic patterns derived from dependency rules were applied. If none of the concepts were available in SenticNet, the *ELM* classifier was employed. The framework outperformed the state-of-the-art statistical methods on movie review dataset.

However, devising machine learning based SA approaches, especially deep models, strongly depends on the experience of researchers. In other words, the design of neural networks lacks theoretical criteria. In 2017, Alain and Bengio [40] studied the intermediate layers of neural networks and drew several insightful conclusions on the basis of accurate experiments. One conclusion from the bifurcation toy model was that even the worse branch could provide complementary information to enhance the performance of another branch. For instance, traditional vector space model and distributed embedding described different aspects of text input, which served as two different branches with complementary information. The second conclusion was that skip connection operation made it easier to train a deep neural network model. Therefore, following the conclusions from that paper, we created a novel SA system entitled lexicon integrated two-channel CNN–BiLSTM model, combining CNN and BiLSTM models in a parallel manner.

2.2. Padding operation

Padding operation is commonly used in natural language processing tasks like sentiment analysis. Input data sample, either a sentence or the whole text in the data sample, is padded with padding element to make the input of a consistent size [55]. One of the frequently used padding operations is zero padding, which fills the missing positions with zero [56,57]. Kim [58] used zero-padding strategy to pad sentences to a fixed length (padded where necessary) for text classification with CNN. Cliche [4] followed the same padding strategy of Kim when conducting SA tasks with both CNN and LSTM. Yu and Chang [59] found that it did not translate well to their dataset when implementing Kim's zero padding strategy. Therefore, they instead segmented each review into chunks of a fixed length and used CNN to predict the sentiment orientation of each one. Feng and Lin [60] used gated recurrent unit to conduct sentiment classification on food reviews. They made use of zero padding to pad the reviews to 88 words and zero padding was done at the beginning instead of the end in their model. It demonstrated that zero padding did benefit the test accuracy of their model. Prusa and Khoshgoftaar [61] introduced a novel text data representation method which represented special characters (characters that are not contained in a pre-defined alphabet) as a vector of zeros. Wang et al. [62] proposed a select-additive learning procedure that improved prediction accuracy significantly in all three modalities (verbal, acoustic, visual) of multimodal SA. For the verbal channel, they concatenated the word embeddings of all the words in the sentence and padded them with appropriate zeros to make them of the equal dimension (fixed length is 60). In addition to zero padding, Song et al. [63] proposed attention-based LSTM model using sentiment lexicon embedding for aspect-level SA in Korean. They introduced sentiment information into word embedding to improve the performance of LSTM based deep learning model. Since the lengths of the reviews are different, they also applied zero padding method to this problem. Munkhdalai et al. [64] introduced a special padding character in order to construct full binary tree for SA.

Zero padding is frequently used in deep learning based SA models. The primary goal of this operation is to ensure that the inputs are of a fixed length. In some cases, zero padding is supposed to benefit the performance of the models, while in other cases, zero padding cannot improve the performance of the models. Besides zero padding, some special characters are also used as padding characters. Based on these observations, sentiment padding is designed to achieve the primary goal and improve the SA performance of deep learning based models.

3. Proposed lexicon integrated two-channel CNN–BiLSTM model

In this section, lexicon integrated two-channel CNN–BiLSTM model is proposed and described in detail. The core contributions of the model consist of two parts, namely sentiment padding and parallel two-channel CNN–BiLSTM structure. Sentiment padding is specially designed for text SA task. Unlike zero padding, sentiment padding diminishes the problem of gradient vanishing between input layer and the first hidden layer. Moreover, zero padding does not refer to "none" in vector space compared with it in grayscale map. Sentiment padding enlarges the proportion of sentiment information in a review, which is useful for sentiment polarity classification. The CNN model is skilled at extracting local features while the BiLSTM model is capable of processing long sequence. In this case, combining CNN with BiLSTM in a parallel manner will provide valuable information for each other.

3.1. Sentiment padding (lexicon integrated features)

One of the biggest challenges for text classification is the variable sequence length. The length of a sentence or message ranges from one word to almost one hundred words in some text datasets like Stanford Sentiment Treebank [65]. It becomes a severe problem especially when CNN model is applied to text classification. In this context, the padding operation, commonly used in image classification to pad the border, inspires us to deal with the aforementioned problem in text classification. Sentiment padding operation introduces sentiment lexicon information into constructed textual features. Such features are called lexicon integrated features in this paper.

A sentence or message is extended into a fixed-length one determined by a few heuristic rules through padding operation. However, unlike the image classification, zero padding [55] is not the best choice for text classification. Zero stands for the minimum of gray-scale map and is used to fill vacancies [66,67], which results from convolution operation at each image's border. Zero padding is reasonable for images because the vacancies in images are of small amounts and have little influence on convolution results. However, in text classification, the lengths of sentences change frequently. In this case, zero padding leads to large quantities of invalid information and reduce the performance of classifiers. Additionally, it has a large effect on the result of LSTM and CNN family model in text classification, since it influences the process of pooling and weight updating. For instance, if there are two sentences with different sentiment polarities that contain only one word and two respectively, zero padding operation makes it scarcely possible to classify these two sentences into correct sentiment polarities. Thus, we propose a novel padding operation named sentiment padding, which extends a sentence or message to a fixed length to better match text classification tasks.

Given an input sentence or message $S = (w_1, w_2, \ldots, w_n)$ that contains n words, it can be transformed into a feature vector $\mathbf{x}_r = (\vec{a_1}, \vec{a_2}, \ldots, \vec{a_n})$, where $\vec{a_i}$ stands for the word embedding of the corresponding word w_i. A generated dictionary stores the word embedding of each word w_i from a certain sentence or message. The input contains various articles, prepositions and conjunctions, that might play no role. Moreover, some nouns, pronouns, even adjectives do not convey any sentiments. Thus, we introduce the sentiment lexicon [68] to estimate whether w_i would belong to sentiment words. Sentiment words benefit the extraction of the sentiment information in S. If the length of feature vector $\mathbf{x}_r$ is less than the fixed length, it is extended according to the following formula:

$$\mathbf{x}_g = (\vec{a_{s_1}}, \vec{a_{s_2}}, \ldots, \vec{a_{s_m}}) \tag{1}$$

$$m = \text{number of sentiment words in } \mathbf{x}_r \tag{2}$$

$$N_i = \left[\frac{|senti\_s_i|}{\sum_{j=1}^{m} |senti\_s_j|} \times p \right] \quad for\ i \in [1, m-1] \cap \mathbb{N} \tag{3}$$

$$N_m = p - \sum_{i=1}^{m-1} N_i \tag{4}$$

$$\mathbf{x}_p = (\vec{a_{n+1}}, \vec{a_{n+2}}, \ldots, \vec{a_{n+p}}) \tag{5}$$

$$\mathbf{x} = (\mathbf{x}_r, \mathbf{x}_p) \tag{6}$$

Under this condition, sentiment lexicon is applied to the extraction of sentiment words from $\mathbf{x}_r$. $\mathbf{x}_g$ consists of the word vectors of these extracted sentiment words, which is sorted in an ascending order by the absolute value of sentiment scores. However, there are p positions between $\mathbf{x}_r$ and the fixed length. In order to embed these m sentiment words into p positions, N_i is set to decide the number of positions that each sentiment word occupies, where $senti\_s_i$ represents the sentiment score given by the existing sentiment lexicon[1] of the *ith* sentiment word in $\mathbf{x}_g$. N_i is aimed at measuring the relative importance of each sentiment word in $\mathbf{x}_g$. It means the sentiment words with strong sentiments are supposed to occupy more positions. Fig. 1 shows the process of sentiment padding when $m < p$. After embedding these m sentiment words into p positions, we obtain vector $\mathbf{x}_p$, where each position in $\mathbf{x}_p$ is a word vector from $\mathbf{x}_g$. Based on the embedding formulas (3) and (4), different positions may hold the same word vector in $\mathbf{x}_g$. Next, the feature vector $\mathbf{x}_r$ and padding vector $\mathbf{x}_p$ are concatenated into a single vector $\mathbf{x}$. For example, "*A truly moving experience, and a perfect example of how art – when done right – can help heal, clarify, and comfort.*" In this sentence, "clarify", "comfort", "moving", "heal", "perfect", and "right" are sentiment words with sentiment scores 0.046, 0.246, 0.623, 0.861, 0.867, and 0.926, respectively. The fixed length in this example is set as 30. Thus, $p = 10$. In these words, the sentiment word with the highest sentiment score is "right" (0.926). And N_i of other sentiment words are 0, 1, 2, 2, and 2, respectively. The N_i of "right" is $p - 7 = 3$. The sentence after sentiment padding is shown as follows: "*A truly moving experience and a perfect example of how art when done right can help heal clarify and comfort comfort moving moving heal heal perfect perfect right right right*".

On the other hand, if the length of feature vector $\mathbf{x}_r$ exceeds the fixed length ($m \geq p$), we first utilize a list of stopwords to remove irrelevant words. In this case, the majority of samples

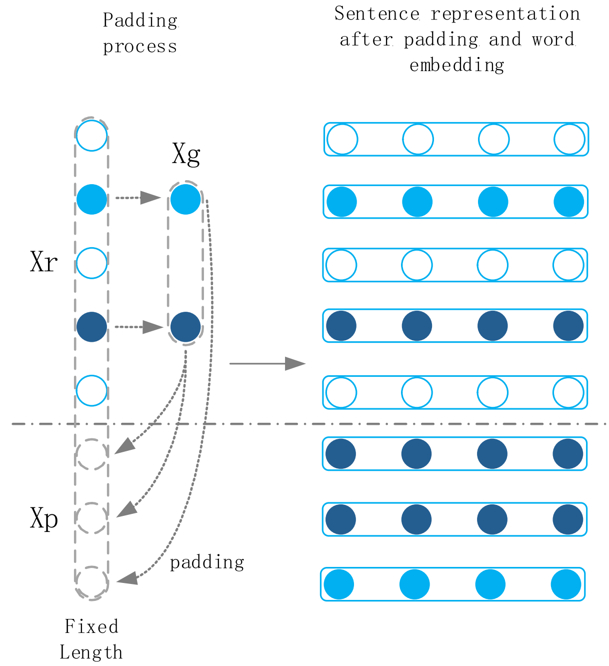

Fig. 1. Brief example of the proposed padding algorithm.[2]

are less than the fixed length, then they are further padded using the aforementioned padding algorithm. For the samples that are still greater than the fixed length, we filter the low-frequency words until it reaches the fixed length. Since the relative number of sentiment words increases, the final vector $\mathbf{x}$ enlarges the influence of sentiment words and reduces the interference of non-sentiment words.

Sentiment padding operation augments the difference between sentences in some cases and generates integrated high-quality lexicon features for SA. Besides, the fixed length vector promotes the application of some machine learning approaches like SVM and CNN.

3.2. The architecture of proposed two-channel model

Alain and Bengio [40] gave some insightful conclusions regarding devising and training deep models. They put forward a hypothetical example in which a model contains a bifurcation which recombines subsequently (Fig. 2). Experimental results demonstrated that even if the performance of one branch is poor, it may also provide complementary information reducing the prediction error of the other branch in certain cases. Skip connection refers to adding input information to the middle layer of deep models (Fig. 3). Researchers found that skip connection operation significantly promoted the training process of deep models. Motivated by these conclusions, we design a two-channel CNN–BiLSTM model incorporating CNN and BiLSTM into a framework in a parallel manner.

Text, like tourism reviews, is a kind of unstructured and heterogeneous data, which means it is hard for the computer to directly process such data. One-hot representation is commonly used to encode text data as input for machine learning based approaches. Since the release of distributed representation Word2vec [69] and GloVe [70], the performance of machine learning based approaches, especially deep models, improved

[1] In this paper, we use NLTK to score English sentiment words and employ domain-specific sentiment lexicon to score Chinese sentiment words.

[2] Fig. 1 illustrated the condition that the length of feature vector x_r is less than the fixed length.

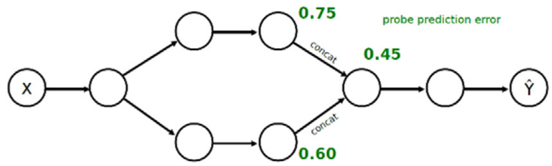

Fig. 2. Architecture of bifurcating toy model [40].

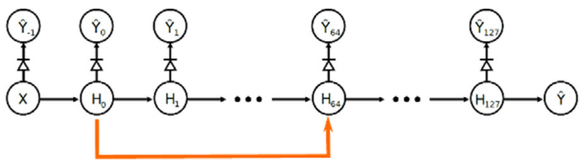

Fig. 3. Illustration of skip connection operation [40].

dramatically. Word2vec is skilled in measuring the semantic information, and GloVe takes both the local and global information into consideration by exploiting statistical methods. LSTM model has been proven to be able to store textual order information and CNN model has been shown to be capable of extracting local features of the text. In this context, it is rational to associate these two models in a parallel manner. Moreover, skip connection operation is added to the intermediate hidden layer of the proposed model.

Fig. 4 shows the architecture of the proposed two-channel CNN–BiLSTM model (without skip connection). The first channel is the BiLSTM channel, where word2vec is used for word embedding. It is composed of one hidden layer and one full connection layer. The second channel is the CNN channel and its input is a single-channel "image" comprising of word2vec embeddings of all the words in a sentence or message. CNN channel consists of six layers, namely two convolutional layers, two chunk-max pooling layers, and two full connection layers. Finally, the outputs of the two-channel CNN and BiLSTM are recombined. In [40], concatenation operation is applied to combining the outputs of two branches. However, add operation is also available for the combination of two channels. Concatenation operation needs an additional full connection layer, which increases the number of parameters. Thus, add operation is chosen to combine the outputs in this article and is formulated in formula (7), where $\mathbf{y}_{bilstm}$ and $\mathbf{y}_{cnn}$ are the outputs of BiLSTM and CNN channels, respectively. The weight λ is set to be a fixed number to determine the relative importance of these two branches. Next, the integrated output $\mathbf{y}$ is mapped into probability $f(\mathbf{y})$ through softmax function and further added into the loss function. In this paper, cross-entropy is chosen as the optimization function. Some notations of formula (8) are introduced first. C is the number of classes and n represents batch size. t_{ki} stands for the component of tag vector of the kth sample. If $t_{ki} = 1$, it means the kth sample belongs to the ith class. $f(y_{ki})$ corresponds to the probability that the kth sample is classified into the ith class using the integrated output y_{ki}, while $f(y_{ki}^{cnn})$ represents the probability that the kth sample is classified into the ith class using the output y_{ki}^{cnn} of CNN channel. Adding cross-entropy loss of CNN channel to the total loss function in formula (8) makes it possible to use different learning rates when training the two channel model. $L_2(\mathbf{w})$ is the L2 regularization dealing with overfitting.

$$\mathbf{y} = \lambda \mathbf{y}_{lstm} + (1-\lambda)\mathbf{y}_{cnn} \tag{7}$$

$$L = -\sum_{k=1}^{n}\sum_{i=1}^{C} t_{ki}(\log(f(y_{ki})) + \log(f(y_{ki}^{cnn}))) + L_2(\mathbf{w}) \tag{8}$$

Table 1
Statistics of SST dataset.

Label	Train	Dev	Test	Sum
1	1092	139	279	1510
2	2218	289	633	3140
3	1624	229	389	2242
4	2322	279	510	3111
5	1288	165	399	1852
Sum	8544	1101	2210	11855

Table 2
Statistics of Chinese Tourism Review dataset.

Label	Train	Dev	Test	Sum
0 (*Negative*)	2893	700	401	3994
1 (*Positive*)	2893	700	401	3994
Sum	5786	1400	802	7988

Besides, we also propose another variant of lexicon integrated two-channel model named CNN–LSTM. The architecture of two-channel CNN–LSTM model is the same as CNN–BiLSTM model in this paper except that the BiLSTM layer in Fig. 4 is replaced by a LSTM layer. The difference between these two models is that BiLSTM layer makes use of both forward and backward information, which contributes to the performance improvement of SA. Nevertheless, these two models have its own strengths in dealing with different SA tasks.

4. Experiments

In this section, we conduct experiments on benchmark datasets to examine the performance of the proposed CNN–LSTM and CNN–BiLSTM models, the coupling of two-channel models, the influence of sentiment padding, and the function of skip connections. Besides, we also analyze the value of parameter λ. The accuracies in the following tables are expressed as a percentage.

4.1. Corpora

The proposed lexicon integrated two-channel CNN–LSTM and CNN–BiLSTM models, some other derivative models, and state-of-the-art approaches are conducted on two SA datasets to evaluate their performances. The first dataset is Stanford Sentiment Treebank (*SST*) [65] which is a famous open SA dataset composed of movie reviews. The following sentence is an example from SST dataset. "Part of the charm of Satin Rouge is that it avoids the obvious with humor and lightness". Besides, we construct another dataset named reversed SST through reversing dev set and test set in SST. Another one is the Chinese tourism review dataset. The following sentence is an example from this dataset. "卖家的服务太棒了,考虑非常周到， 完全超出期望值。玩得真的很开心。很满意的旅游。" (The service is very good. Well arranged journey. The service is beyond our expectation. I am really have fun. Very satisfied with the trip.) Detailed statistical information of SST and Chinese tourism review datasets are listed in Tables 1–2. Labels 1–5 in Table 1 correspond to very negative, negative, neutral, positive, and very positive, respectively.

The Chinese dataset consists of tourism reviews from online travel websites Ctrip.com[2] and Qunar.com.[3] We invited fifteen students to label these tourism reviews and each review is annotated by three students, ensuring the quality of the Chinese tourism review dataset. Each review contains several sentences describing different aspects of an online travel product.

2 http://www.ctrip.com.
3 https://www.qunar.com.

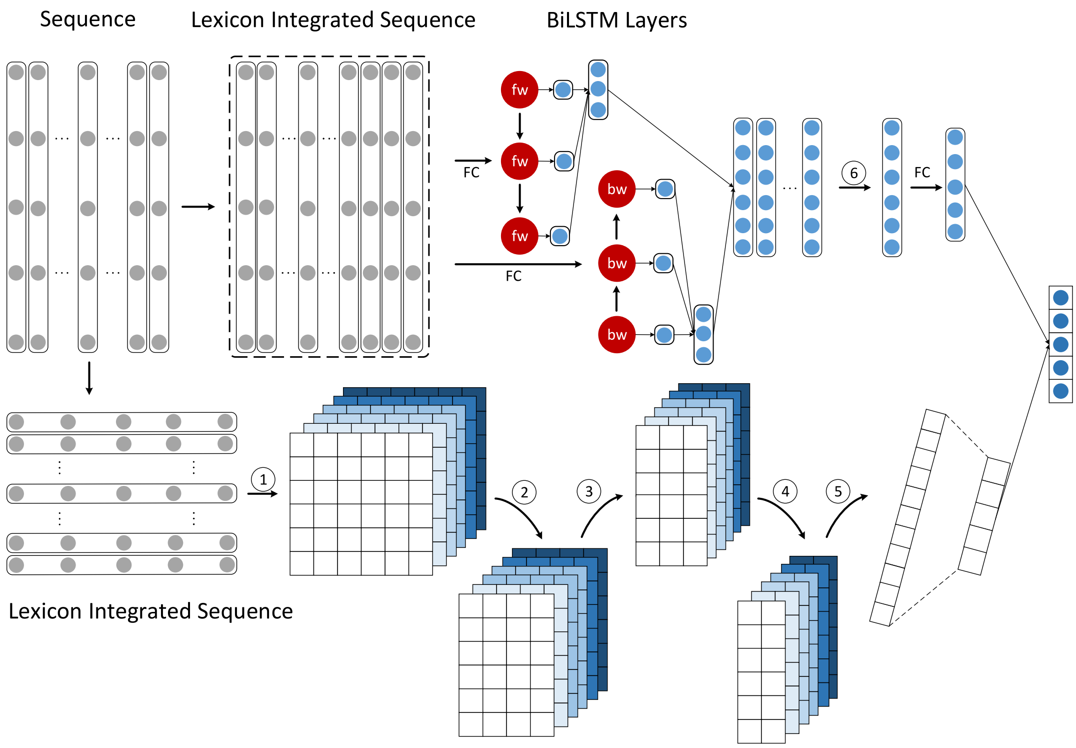

Fig. 4. System architecture of lexicon integrated two-channel CNN–BiLSTM model. 1: First convolutional layer; 2: Chunk-max pooling layer; 3: Second convolutional layer; 4: Second chunk-max pooling layer. Note: In BiLSTM layers, "fw" node refers to forward node, and "bw" node refers to backward node. FC means full connection. Skip connection operation is not illustrated in this figure.

For example, a typical tourism review is supposed to depict the itinerary, cicerone, weather and restaurant of an online travel service. Stanford Sentiment Treebank (SST) dataset is a standard English movie review dataset on which many state-of-the-art SA models evaluate its performances. Almost all the reviews in this dataset consist of a single sentence. SST is a fine-grained movie review dataset and Chinese tourism review dataset is a coarse-grained tourism review dataset.

4.2. Baselines and parameter set

The baseline models include Support Vector Machine (SVM) [71], Nonparallel Support Vector Ordinal Regression (NSVOR) [71], Neural Network (NN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Long Short-Term Memory Model (LSTM), Bidirectional Long Short-Term Memory Model (BiLSTM), Tree-Structure LSTM (TreeLSTM) [33], state-of-the-art sentiment lexicons [72] and some other derivative models. Among these models, the parameters of logistic regression (primal, dual), support vector machine (primal, dual, Crammer and Singer) and support vector ordinal regression are the same as they are in [71]. The number of hidden layers and nodes of neural networks and recurrent neural networks are the same as those of LSTM channel in this paper. CNN†, LSTM† and BiLSTM† are the derivative models originating from one branch of the proposed CNN–LSTM or CNN–BiLSTM model. Among the aforementioned baselines, TreeLSTM is the state-of-the-art SA method, especially on the SST dataset. Word representations used in this paper are Word2vec [69] and GloVe [70] (200 dimensions).

In this paper, the fixed length are set to be 30, 30 and 70 for SST, reversed SST and Chinese tourism review dataset, respectively.[4] The initial learning rate is set to 0.1 which decays by 0.05 every 300 training steps. The batch size is set to 25. For the BiLSTM branch, there is one hidden layer with 200 BiLSTM cells. For the CNN branch, there are two convolutional layers. The first layer is composed of 32 kernels with the shape 3×5. The second layer consists of 32 kernels with the shape 3×3. Cross entropy loss function is utilized to measure the training of the model and gradient descent algorithm was applied to searching for the optimal parameters. The initializer used in our models is Xavier (uniform mode).

During the implementation process, we first use sentiment padding operation to process raw data into lexicon integrated features. Then these features are fed to two different channels at the same time, one is CNN channel, and the other is BiLSTM/LSTM channel. Next, the results from two channels are concatenated and fed to a neural network classifier (as is shown in Fig. 4).

4.3. Sentiment classification experiment

In this subsection, we conduct three groups of experiments on SST dataset, reversed SST dataset, and Chinese tourism review dataset, respectively. In Tables 3–5, our proposed methods are in bold and the methods in italic are the methods whose results are reported by the literature. Other methods in Tables 3–5 are the ones of which the results are obtained from the experiments conducted by us. For our experiments, we report the mean accuracy over 3 runs.

As is shown in Table 3, the experiments performed on the SST dataset utilize approved machine learning approaches like logistic regression and support vector ordinal regression [76], traditional sentiment lexicon SentiWordNet [77], and advanced neural network models. We apply word2vec [69] and glove [70] to represent unstructured text data. Word embedding models are trained on a large corpus composed of amazon_mp3 product reviews, IMDB reviews, TripAdvisor reviews,[5] etc.

[4] The fixed length is set according to heuristic rule that almost eighty percent of the samples are less than the fixed length.

[5] http://www.cs.virginia.edu/~hw5x/dataset.html.

Table 3
Sentiment classification accuracies on SST dataset.

Methods	Representation	Fine-grained
Logistic Regression(primal) [71]	LibSVM Format	39.7285
Logistic Regression(dual) [71]	LibSVM Format	39.8190
Support Vector Machine(primal) [71]	LibSVM Format	39.9095
Support Vector Machine(dual) [71]	LibSVM Format	39.9548
Support Vector Machine(Crammer and Singer) [71]	LibSVM Format	37.5566
Support Vector Ordinal Regression [71]	LibSVM Format	34.2986
RAE [65]	–	43.2
MV-RNN [65]	–	44.4
RNTN [65]	–	45.7
DCNN [73]	–	48.5
Paragraph-Vec [74]	–	48.7
CNN-non-static [58]	–	48.0
CNN-multichannel [58]	–	47.4
DRNN [75]	–	49.8
LSTM [33]	Glove.840B.300d[a]	46.4
Bidirectional LSTM [33]	Glove.840B.300d	49.1
2-layer LSTM [33]	Glove.840B.300d	46.0
2-layer Bidirectional LSTM [33]	Glove.840B.300d	48.5
Dependency Tree-LSTM [33]	GloVe.840B.300d	48.4
Constituency Tree-LSTM [33]	GloVe.840B.300d, fixed	49.7
Constituency Tree-LSTM [33]	GloVe.840B.300d, tuned	51.0
CNN–RNN [51]	Word2vec	**51.5**
Neural Network	One-hot	29.7285
	Word2vec	33.4841
	GloVe	30.9049
Recurrent Neural Network	Woed2vec	41.8
LSTM [33]	Word2vec	42.76
Bidirectional LSTM [33]	Word2vec	42.40
2-layer LSTM [33]	GloVe	43.08
2-layer LSTM [33]	Word2vec	43.3
2-layer Bidirectional LSTM [33]	GloVe	45.75
2-layer Bidirectional LSTM [33]	Word2vec	44.39
Dependency Tree-LSTM [33]	Glove	43.89
Dependency Tree-LSTM [33]	Word2vec	42.99
Constituency Tree-LSTM [33]	Glove	46.83
Constituency Tree-LSTM [33]	Word2vec	47.51
Long Short-term Memory Model†	Word2vec	44.6003
Bidirectional LSTM†	Word2vec	45.4389
Convolutional Neural Network†	Word2vec	42.368
CNN–LSTM (our sentiment padding)	Word2vec	47.9126
CNN–BiLSTM (zero padding [50]**)**	Word2vec	47.5566
CNN–BiLSTM (our sentiment padding)	Word2vec	49.8944[b]

[a]GloVe.840B.300d is a pre-trained word embedding by Stanford NLP group. https://nlp.stanford.edu/projects/glove/.
[b]The best results are in bold.

Neither the sentiment classification accuracy of the CNN branch nor the LSTM branch exceeds 45 percent. However, when these two branches are incorporated into a CNN–LSTM model, the performance is close to 48 percent. It proves that the parallel two-channel structure takes advantage of the features extracted from both channels comprehensively. Furthermore, we use BiLSTM model to replace the LSTM branch, the performance of the CNN–BiLSTM model outperforms all the benchmark models except the Constituency Tree-LSTM and CNN–RNN model created by Wang et al. [51]. However, it is time-consuming to parse sentences especially Chinese sentences for TreeLSTM model.[6] In this case, the CNN–BiLSTM model has its unique strength. Compared to LSTM model, BiLSTM model does well in extracting features. In contrast with zero padding, our sentiment padding gets better results on SA task, indicating the effectiveness and superiority of this padding operation.

Table 4 shows the SA results on the reversed SST dataset. The reversed dataset is obtained by swapping the dev and test sets in *SST* dataset. In this context, the combination of CNN and LSTM branches improves the accuracy of sentiment classification by a substantial margin. The performance gain on reversed SST dataset is more obvious than on SST dataset. This coincides with the conclusion in [40] that one of the branches may contain very meaningful features which the other branch may not have.

In this paper, we utilize the new words detection algorithm [78] to do the Chinese text segmentation. Table 5 demonstrates that, compared with several approved machine learning models, sentiment lexicon and neural network models, the proposed model shows the superiority in detecting the sentiment polarities of Chinese tourism reviews.

Summarizing the results in Tables 3–5, we conclude that the proposed two-channel models show superiority on both English Movie Review and Chinese Tourism Review SA tasks.

In Tables 3 and 4, paralleled two-channel models have unique strength compared with any single branch model (CNN†, LSTM† or BiLSTM†) on English SA dataset. However, it should be noted that, in Table 5, the performance of CNN–BiLSTM model is poorer than its BiLSTM branch. In addition, in Tables 3 and 5, the performance of BiLSTM model is better than LSTM model. Nevertheless, this does not mean CNN–BiLSTM must be better than CNN–LSTM. For example, in Table 3, the accuracy of CNN–LSTM is 47.91 percent and the accuracy of *CNN–BiLSTM* is 49.89 percent while in Table 5, the accuracy of CNN–BiLSTM is 3.76 percent lower than that of CNN–LSTM. Furthermore, the accuracy of CNN–BiLSTM is even lower than that of BiLSTM in Table 5. This phenomenon indicates that the performance of parallel two-channel model does

[6] It takes about 4.5 h to parse 1000 lines Chinese tourism reviews (Intel (R) Core(TM) i7-6700HQ CPU @ 2.60 GHz)

Table 4
Sentiment classification accuracies on SST dataset(reversed).

Methods	Representation	Fine-grained
Logistic Regression(primal) [71]	LibSVM Format	38.0563
Logistic Regression(dual) [71]	LibSVM Format	38.0563
Support Vector Machine(primal) [71]	LibSVM Format	37.1480
Support Vector Machine(dual) [71]	LibSVM Format	37.2389
Support Vector Machine(Crammer and Singer) [71]	LibSVM Format	36.5520
Support Vector Ordinal Regression [71]	LibSVM Format	37.5114
Neural Network	One-hot	30.0640
	Word2vec	26.249
	GloVe	29.973
Recurrent Neural Network	Word2vec	48.1
Long Short-term Memory Model†	Word2vec	43.5967
Convolutional Neural Network†	Word2vec	42.4160
CNN–LSTM (our sentiment padding)	Word2vec	**50.6812**
CNN–BiLSTM (our sentiment padding)	Word2vec	50.2271

Table 5
Sentiment classification accuracies on Chinese Tourism dataset.

Methods	Representation	Binary
Logistic Regression(primal) [71]	LibSVM Format	57.1072
Logistic Regression(dual) [71]	LibSVM Format	57.1072
Support Vector Machine(primal) [71]	LibSVM Format	56.4838
Support Vector Machine(dual) [71]	LibSVM Format	55.8603
Support Vector Machine(Crammer and Singer) [71]	LibSVM Format	65.8354
Support Vector Ordinal Regression [71]	LibSVM Format	65.9601
Neural Network	One-hot	84.788
	Word2vec_C	92.6434
	GloVe_C	92.0199
SentiRUC [72]	–	66.8329
Recurrent Neural Network	Word2vec_C	92.5187
Recurrent Neural Network	GloVe_C	91.3965
Long Short-term Memory Model†	Word2vec_C	89.2768
Long Short-term Memory Model†	GloVe_C	92.2693
Convolutional Neural Network†	Word2vec_C	88.7781
Convolutional Neural Network†	GloVe_C	92.0202
Bidirectional LSTM†	Word2vec_C	91.8333
CNN–LSTM (our sentiment padding)	Word2vec_C	**95.0125**
CNN–LSTM (our sentiment padding)	GloVe_C	93.3915
CNN–BiLSTM (our sentiment padding)	Word2vec_C	91.2500

not necessarily depend on the performance of any single branch; instead, it depends on the coupling between two branches. Taking Table 5 as an example, the coupling between CNN and LSTM branches is better than that between CNN and BiLSTM branches. One possible reason is that the strength of coupling is related to corpora.

4.4. Analysis of sentiment padding and skip connection

Tables 6 and 7 show the influence of the lexicon integrated features on reversed *SST* dataset and Chinese tourism review dataset. Apparently, sentiment lexicon and padding operation contribute to the improvements of SA accuracy in most cases. The parameter *fixed sequence length* is devised according to the experience and there is a heuristic rule to determine the exact value of the *fixed sequence length*. Motivated by the Pareto law of the vital few and trivial many, a rule is set where the length of eighty percent of the sentences or messages should be less than the chosen *fixed length*. Thus this *fixed length* can be determined by the cumulative distribution function of the sentence or message length in the related dataset. Besides our proposed heuristic rule, there are some other methods determining the fixed sequence length. Feng and Lin [60] used the average length of reviews as fixed length. Wang et al. [62] set the fixed length as 60 and discarded additional words. In this case, only around 0.5% reviews in their datasets have more than 60 words. The first and the second methods are suitable for the case that the variance of review length is relatively low. However, the second method may need many padding characters.

Table 6
Statistics of the influence of sentiment padding (reversed SST dataset).

Methods	Representation	Sentiment padding	Zero padding
CNN†	Word2vec	42.4160	41.4169
RNN	Word2vec	48.1198	34.9177
LSTM†	Word2vec	43.5967	40.9628
BiLSTM†	Word2vec	45.4389	44.5249
CNN–LSTM	Word2vec	50.6812	47.0481

Table 7
Statistics results of the influence of sentiment padding (Chinese tourism review dataset).

Methods	Representation	Sentiment padding	Zero padding
CNN†	Word2vec_C[a]	88.7781	88.5287
CNN†	GloVe_C	92.0202	91.3965
RNN	Word2vec_C	92.5187	92.2693
RNN	GloVe_C	91.3965	89.2768
LSTM†	Word2vec_C	89.2768	84.6633
LSTM†	GloVe_C	92.2693	90.5237
CNN–LSTM	Word2vec_C	95.0125	94.6384
CNN–LSTM	GloVe_C	93.3915	93.5162

[a]Word2vec_C and GloVe_C are the Chinese word embeddings trained on Zhiwiki corpus through word2vec and glove model individually. http://download.wikipedia.com/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2.

Table 8
Statistics results of the influence of skip connection.

Skip Input	Skip connection(100%)	Skip connection (10%)[a]	Without skip connection
Senti_input	44.7775	40.1453	50.6812
Word2vec	42.5068	41.0536	50.6812
GloVe	39.6004	41.4169	50.6812

[a]It means that the input of skip connection operation is the original input information multiplied by 0.1.

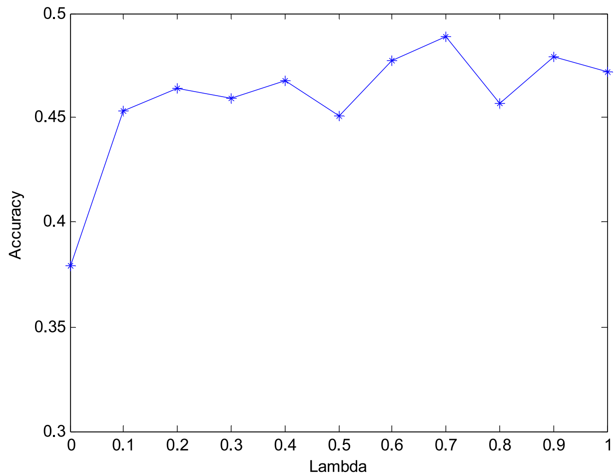

Fig. 5. Parameter analysis of λ.

Analysis of the influence of skip connection is listed in Table 8. Skip input stands for the type of input added into a certain hidden layer, where senti_input is lexicon integrated feature; Word2Vec and GloVe represent word embedding features. It indicates that skip connection operation decreases the accuracy of sentiment classification without speeding up the training process. Thus we conclude that skip connection operation is not suitable for training neural network models proposed in this paper. Perhaps two reasons contribute to this phenomenon. First, skip connection is suitable for deep learning model with many hidden layers to provide important initial information in case of forgetting. But the proposed two channel model is not deep enough to forget the initial raw information. Second, LSTM family is qualified for processing long-term information, and this will lower the demand for skip connection to provide 'reminder'. Therefore, skip connection may bring too much attention to the raw information and lower the performance of the proposed model.

4.5. Parameter analysis of λ

As is shown in formula (7), the weight λ measures the relative importance of two different branches, which has a huge influence on the final results. Therefore, in this subsection, the influence of λ on SA is discussed. All the experiments are conducted on Stanford Sentiment Treebank dataset. In Fig. 5, as λ changes from 0 to 0.1, the accuracy of the CNN–BiLSTM model increases about 7 percent. As λ changes from 0.1 to 0.6, it fluctuates around 0.46. Then it reaches the maximum when $\lambda = 0.7$. As λ increases from 0.8 to 1.0, the accuracy fluctuates around 0.47. In general, moderate value of λ ranges from 0.6 to 0.9 according to the parameter analysis result. This can be explained in a more practical sense that BiLSTM channel, mainly considering the sequential information, has more useful information in comparison with the CNN channel. The exact optimal value is 0.7 in our CNN–BiLSTM model. However, the value may be different for other parallel two-channel models since the relative importance of different channels varies a lot.

5. Conclusion

In this paper, we proposed the lexicon integrated two-channel CNN–LSTM and CNN–BiLSTM models. Taking into consideration the sentiment lexicon information, we first proposed sentiment padding method to make the input data sample of a consistent size and improve the proportion of sentiment information in each review. Sentiment padding alleviated the problem of gradient vanishing between the input layer and the first hidden layer, which may appear while using zero padding. Sentiment padding operation generated integrated high-quality lexicon features for SA. Meanwhile, experiments on several SA datasets demonstrated the superiority of sentiment padding. Motivated by the bifurcating toy model and skip connection operation [40], we presented the two-channel CNN–BiLSTM model combining the CNN and BiLSTM branches in a parallel manner. Well-designed loss function made it possible for the two branches to train in different paces. The parallel two-channel CNN–BiLSTM model improved the SA performances on *SST* and reversed SST datasets by a large margin. However, we found that CNN–LSTM model got better performance on Chinese tourism review dataset compared with CNN–BiLSTM model. Moreover, CNN–BiLSTM model even could not exceed its single branch BiLSTM on Chinese tourism dataset. This indicates that the performance of parallel two-channel model does not necessarily depend on the performance of its single branch; instead, it depends on the coupling between two branches. Corpora may influence the coupling between two branches.

In addition, lexicon integrated two-channel CNN–LSTM model obtained the best result on the Chinese tourism review dataset. Extensive experiments demonstrated that sentiment lexicon information and parallel two-channel model contributed to the significant improvements of SA accuracy. In addition, it has been proved that skip connection operation could not improve the performance or speed up the training process of neural network models proposed in this paper.

We hope that the design of the proposed model following the theoretical guidance benefits the understanding of deep learning models and provide novel perspectives for SA. In the future study, the factor that influences the coupling between two branches should be the research focus.

CRediT authorship contribution statement

Wei Li: Software, Methodology, Conceptualization, Investigation, Validation, Writing - original draft. **Luyao Zhu:** Conceptualization, Project administration, Visualization, Writing - review & editing. **Yong Shi:** Supervision, Funding acquisition. **Kun Guo:** Supervision, Resources. **Erik Cambria:** Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046), the National Natural Science Foundation of China No. 91546201, No. 71331005 and No. 71501175. We would like to thank for the anonymous reviewers.

References

[1] P. Nakov, A. Ritter, S. Rosenthal, V. Stoyanov, F. Sebastiani, SemEval-2016 Task4: Sentiment analysis in Twitter, in: Proc. 10th Int. Work. Semant. Eval., 2016, pp. 1–18.
[2] F. Wu, Y. Huang, Y. Song, S. Liu, Towards building a high-quality microblog-specific chinese sentiment lexicon, Decis. Support Syst. 87 (2016) 39–49, http://dx.doi.org/10.1016/j.dss.2016.04.007.
[3] R. Socher, B. Huval, C.D. Manning, A.Y. Ng, Semantic compositionality through recursive matrix-vector spaces, in: Proc. 2012 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn., 2012, pp. 1201–1211.
[4] M. Cliche, BB Twtr at semeval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs, 2017.
[5] A.A. Al Sallab, R. Baly, H. Hajj, Deep learning models for sentiment analysis in arabic, 2015, pp. 9–17.
[6] A. Perelygin, Semantic compositionality and deep learning for sentiment analysis, 2012, pp. 1–9.
[7] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Twenty-Ninth AAAI Conf. Artif. Intell., 2015, pp. 2267–227.
[8] Y. Yu, H. Lin, J. Meng, Z. Zhao, Visual and textual sentiment analysis of a microblog using deep convolutional neural networks, Algorithms 9 (2016) 41, http://dx.doi.org/10.3390/a9020041.
[9] X. Ouyang, P. Zhou, C.H. Li, L. Liu, Sentiment analysis using convolutional neural network, in: Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Auton. Secur. Comput. Pervasive Intell. Comput. (CIT/IUCC/DASC/PICOM), 2015 IEEE Int. Conf., 2015, pp. 2359–2364, http://dx.doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.349.
[10] Z. Toh, J. Su, NLANGP at SemEval-2016 task 5: Improving aspect based sentiment analysis using neural network features, in: Proc. 10th Int. Work. Semant. Eval., SemEval-2016, 2016, pp. 282–288.
[11] S. Poria, E. Cambria, A. Gelbukh, Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis, in: Proc. 2015 Conf. Empir. Methods Nat. Lang. Process., 2015, pp. 2539–2544.
[12] S. Poria, E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowl. Based Syst. 108 (2016) 42–49, http://dx.doi.org/10.1016/j.knosys.2016.06.009.
[13] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proc. 2015 Conf. Empir. Methods Nat. Lang. Process., 2015, pp. 1422–1432.
[14] M. Huang, B. Ye, Y. Wang, H. Chen, J. Cheng, X. Zhu, New word detection for sentiment analysis, acl, 2014, pp. 531–541.
[15] F.H. Khan, U. Qamar, S. Bashir, eSAP: A decision support framework for enhanced sentiment analysis and polarity classification, Inf. Sci. (Ny). 367–368 (2016) 862–873, http://dx.doi.org/10.1016/j.ins.2016.07.028.
[16] W. Medhat, A. Hassan, H. Korashy, Sentiment analysis algorithms and applications: A survey, Ain Shams Eng. J. 5 (2014) 1093–1113, http://dx.doi.org/10.1016/j.asej.2014.04.011.
[17] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K.J. Miller, Introduction to wordnet: An on-line lexical database, Int. J. Lexicogr. 3 (1990) 235–244, http://dx.doi.org/10.1093/ijl/3.4.235.
[18] C. Strapparava, A. Valitutti, WordNet-Affect: an affective extension of WordNet, in: Proc. 4th Int. Conf. Lang. Resour. Eval., 2004, pp. 1083–1086.
[19] H. Cho, S. Kim, J. Lee, J.S. Lee, Data-driven integration of multiple sentiment dictionaries for lexicon-based sentiment classification of product reviews, Knowl. Based Syst. 71 (2014) 61–71, http://dx.doi.org/10.1016/j.knosys.2014.06.001.
[20] F. Bravo-Marquez, E. Frank, B. Pfahringer, Building a twitter opinion lexicon from automatically-annotated tweets, Knowl. Based Syst. 108 (2016) 65–78, http://dx.doi.org/10.1016/j.knosys.2016.05.018.
[21] S.D. Richardson, W.B. Dolan, L. Vanderwende, MindNet: acquiring and structuring semantic information from text, in: COLING-ACL'98 Meet. Assoc. Comput. Linguist., 1998, pp. 1098–1102, http://dx.doi.org/10.3115/980691.980749.
[22] L. Ku, Y. Liang, H. Chen, K. Lun-Wei, L. Yu-Ting, C. Hsin-Hsi, Opinion extraction, summarization and tracking in news and blog corpora, Artificial Intelligence (2006) 100–107.
[23] H. Xu, K. Zhao, L. Qiu, C. Hu, Expanding Chinese sentiment dictionaries from large scale unlabeled corpus, in: Proc. PACLIC 24, 2010, pp. 301–310.
[24] L. Velikovich, S.B. Kerry, H. Ryan, The viability of web-derived polarity lexicons, in: Hum. Lang. Technol. 2010 Annu. Conf. North Am. Chapter Assoc. Comput. Linguist., 2010, pp. 777–785, http://dx.doi.org/10.1056/NEJM199902113400601.
[25] D. Tang, F. Wei, B. Qin, M. Zhou, T. Liu, Building large-scale twitter-specific sentiment lexicon: a representation learning approach, in: Proc. 25th Int. Conf. Comput. Linguist., COLING 2014, 2014, pp. 172–182.
[26] H. Saif, M. Fernandez, Contextual semantics for sentiment analysis of twitter Categories and subject descriptors, Inf. Process. Manag. 52 (2016) 5–19, http://dx.doi.org/10.1016/j.ipm.2015.01.005.
[27] S. Mohammad, C. Dunne, B. Dorr, Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus, in: EMNLP '09 Proc. 2009 Conf. Empir. Methods Nat. Lang. Process., vol. 2, 2009, pp. 599–608, http://dx.doi.org/10.3115/1699571.1699591.
[28] S. Feng, L. Wang, W. Xu, D. Wang, G. Yu, Unsupervised learning chinese sentiment lexicon from massive microblog data, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2012, pp. 27–38, http://dx.doi.org/10.1007/978-3-642-35527-1_3.
[29] F. Wu, Y. Huang, Y. Song, S. Liu, Towards building a high-quality microblog-specific Chinese sentiment lexicon, Decis. Support Syst. 87 (2015) 39–49, http://dx.doi.org/10.1016/j.dss.2016.04.007.
[30] B. Shin, T. Lee, J.D. Choi, Lexicon integrated CNN models with attention for sentiment analysis, 2015.
[31] C.N. dos Santos, M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts, coling-2014, 2014, pp. 69–78..
[32] T. Khalil, Samhaa R. El-Beltagy, NileTMRG: Deep convolutional neural networks for aspect category and sentiment extraction in SemEval-2016 task 5, in: Proc. 10th Int. Work. Semant. Eval., SemEval 2016, 2016, pp. 276–281, in press.
[33] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory networks, in: Proc. ACL, 2015, pp. 1556–1566, http://dx.doi.org/10.1515/popets-2015-0023.
[34] J. Wang, L. Yu, K.R. Lai, X. Zhang, Dimensional sentiment analysis using a regional CNN-LSTM model, 2016, pp. 225–230.
[35] M. Zhang, Y. Zhang, D. Vo, Gated neural networks for targeted sentiment analysis, 2014, pp. 3087–3093.
[36] A.S. Manek, P.D. Shenoy, M. Chandra Mohan, K.R. Venugopal, Aspect term extraction for sentiment analysis in large movie reviews using gini index feature selection method and SVM classifier, in: World Wide Web, Springer, 2016, http://dx.doi.org/10.1007/s11280-015-0381-x.
[37] Y. Fuping, H. Zhiyong, Chinese micro-blog sentiment analysis based on SVM and complex phrasing, 2016, pp. 841–846.
[38] X. Fu, T. Wang, Long short-term memory network over rhetorical structure theory for sentence-level sentiment analysis, 2016, pp. 17–32.
[39] C. Paper, R. Xu, A sequence level latent topic modeling method for sentiment analysis via CNN based diversified restrict Boltzmann machine a sequence level latent topic modeling method for, 2017, http://dx.doi.org/10.1109/ICMLC.2016.7860927.
[40] G. Alain, Y. Bengio, Understanding intermediate layers using linear classifier probes, 2017, pp. 1–11.
[41] F.H. Khan, U. Qamar, S. Bashir, SWIMS: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis, Knowl. Based Syst. 100 (2015) 97–111, http://dx.doi.org/10.1016/j.knosys.2016.02.011.
[42] F.H. Khan, U. Qamar, S. Bashir, Lexicon based semantic detection of sentiments using expected likelihood estimate smoothed odds ratio, Artif. Intell. Rev. (2016) 1–26, http://dx.doi.org/10.1007/s10462-016-9496-4.
[43] S. Minaee, E. Azimi, A. Abdolrashidi, Deep-sentiment: Sentiment analysis using ensemble of CNN and Bi-LSTM models.
[44] M. Rhanoui, M. Mikram, A CNN-BiLSTM model for document-level sentiment analysis, 2019, pp. 832–847, http://dx.doi.org/10.3390/make1030048.
[45] D. Li, J. Qian, Text sentiment analysis based on long short-term memory, 2016, pp. 471–475.
[46] L. Bao, P. Lambert, T. Badia, Attention and lexicon regularized LSTM for aspect-based sentiment analysis, 2019, pp. 253–259.
[47] J. Zhou, Y. Lu, H. Dai, H. Wang, H. Xiao, Sentiment analysis of chinese microblog based on stacked bidirectional LSTM, IEEE Access 7 (2019) 56–66, http://dx.doi.org/10.1109/ACCESS.2019.2905048.
[48] S. Tan, X. Cheng, Y. Wang, H. Xu, Adapting naive bayes to domain adaptation for sentiment analysis, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2009, pp. 362–374, http://dx.doi.org/10.1007/978-3-642-00958-7_33.
[49] B.G. Patra, S. Mandal, D. Das, S. Bandyopadhyay, JU _ CSE: A conditional random field (CRF) based approach to aspect based sentiment analysis, 2014, pp. 370–374.
[50] J. Yoon, H. Kim, Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis, pp. 244–253.
[51] W. Xingyou, W. Jiang, L. Zhiyong, Combination of convolutional and recurrent neural network for sentiment analysis of short texts, 2016, pp. 2428–2437.

[52] V.D. Van, Combining convolution and recursive neural networks for sentiment analysis, 2018, http://dx.doi.org/10.1145/3155133.3155158.
[53] X. Chen, Y. Wang, Q. Liu, Visual and textual sentiment analysis using deep fusion convolutional, 2018.
[54] S. Poria, E. Cambria, G. Winterstein, G. Bin Huang, Sentic patterns: Dependency-based rules for concept-level sentiment analysis, Knowl. Based Syst. 69 (2014) 45–63, http://dx.doi.org/10.1016/j.knosys.2014.05.005.
[55] S. Chintala, Sentiment Analysis using neural architectures, Work. Pap., 2012, pp. 1–4, http://dx.doi.org/10.1016/j.apcatb.2016.03.035.
[56] G. Piao, J.G. Breslin, Financial aspect and sentiment predictions with deep neural networks: an ensemble approach, 2 (2018) 0–4, http://dx.doi.org/10.1145/3184558.3191829.
[57] M. Nabil, M. Aly, A.F. Atiya, CUFE at semeval-2016 task 4: A gated recurrent model for sentiment classification, 2016, pp. 52–57.
[58] Y. Kim, Convolutional neural networks for sentence classification, in: Proc. 2014 Conf. Empir. Methods Nat. Lang. Process., EMNLP 2014, 2014, pp. 1746–1751, http://dx.doi.org/10.1109/LSP.2014.2325781.
[59] A. Yu, D. Chang, Multiclass sentiment prediction using yelp business reviews, CS224n proj, 2011, pp. 2–9.
[60] H. Feng, R. Lin, Sentiment classification of food reviews, 2016, http://arxiv.org/abs/1609.01933.
[61] J.D. Prusa, T.M. Khoshgoftaar, Improving deep neural network design with new text data representations, J. Big Data 4 (2017) http://dx.doi.org/10.1186/s40537-017-0065-8.
[62] H. Wang, A. Meghawat, L. Morency, E.P. Xing, select-additive learning: improving generalization in multimodal sentiment analysis, vol. 1.
[63] M. Song, H. Park, K. Shin, Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean, Inf. Process. Manag. 56 (2019) 637–653, http://dx.doi.org/10.1016/j.ipm.2018.12.005.
[64] T. Munkhdalai, Y. Hong, Neural tree indexers for text understanding, 2017, pp. 11–21, http://dx.doi.org/10.1186/s40945-017-0033-9.Using.
[65] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, 2013.
[66] E. Cambria, D. Hazarika, S. Poria, A. Hussain, Benchmarking multimodal sentiment analysis.
[67] S. Poria, I. Chaturvedi, E. Cambria, Convolutional MKL based multimodal emotion recognition and sentiment analysis, in: 2016 IEEE 16th Int. Conf. Data Min., 2016, pp. 439–448, http://dx.doi.org/10.1109/ICDM.2016.0055.
[68] W.L. Kun Guo, Yong Shi, Luyao Zhu, Yuanchun Zheng, Improved new word detection used in tourism field, Procedia Comput. Sci. 108 (2017) 1251–1260, http://dx.doi.org/10.1016/j.procs.2017.05.022.
[69] T. Mikolov, G. Corrado, K. Chen, J. Dean, Efficient estimation of word representations in vector space, in: Proc. Int. Conf. Learn. Represent., ICLR 2013, 2013, pp. 1–12, http://dx.doi.org/10.1162/153244303322533223.
[70] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proc. 2014 Conf. Empir. Methods Nat. Lang. Process, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162.
[71] H. Wang, Y. Shi, L. Niu, Y. Tian, Nonparallel support vector ordinal regression, IEEE Trans. Cybern. 47 (2017) 3306–3317.
[72] X. Yang, Z. Zhang, Z. Zhang, Y. Mo, L. Li, L. Yu, P. Zhu, Automatic Construction and Global Optimization of a Multi-Sentiment Lexicon, vol. 2016, 2016.
[73] E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, 2014, pp. 655–665.
[74] Q. Le, T. Mikolov, T.G. Com, Distributed Representations of Sentences and Documents, vol. 32, 2014.
[75] C. Cardie, Deep Recursive Neural Networks for Compositionality in Language, 2014, pp. 1–9.
[76] F.B.X. Zhu, Measuring the Non-compositionality of Multiword Expressions, in: Proc. 23rd Int. Conf. Comput. Linguist., Coling 2010, 2010, pp. 116–124.
[77] S. Baccianella, A. Esuli, F. Sebastiani, SentiWordNet, 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in: Proc. Seventh Int. Conf. Lang. Resour. Eval., vol. 0, 2010, pp. 2200–2204.
[78] W. Li, K. Guo, Y. Shi, L. Zhu, Y. Zheng, DWWP: Domain-specific new words detection and word propagation system for sentiment analysis in the tourism domain, Knowl. Based Syst. 146 (2018) 203–214, http://dx.doi.org/10.1016/j.knosys.2018.02.004.