Knowledge-Based Systems 226 (2021) 107135

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Personalized sentiment classification of customer reviews via an interactive attributes attention model[☆]

You Zhang, Jin Wang [*], Xuejie Zhang

School of Information Science and Engineering, Yunnan University, Kunming, China

ARTICLE INFO

Article history:
Received 21 November 2020
Received in revised form 28 February 2021
Accepted 11 May 2021
Available online 12 May 2021

Keywords:
Sentiment analysis
Text classification
Sentence representation
Bilinear attention

ABSTRACT

Incorporating extra attributes of customer reviews, such as user and product information, to align text representations to each attribute has improved the sentiment polarity classification performance. Existing works only treated such attributes separately thus ignored the interactive information between these attributes. In this paper, we proposed an interactive attributes attention model that considered all attributes to be relevant and investigated the interactive relationships in and across separate features to improve the sentiment classification performance for customer reviews. In addition to the local text encoder, three more interactive attribute encoders, including user–product, user–text, and product–text encoders, are applied to extract implicit information to align attribute features to text representations with a bilinear interaction instead of self-attention. To better integrate different information, a multiloss objective function is used to further improve the performance. The comparative experiments on the IMDB, Yelp, and Amazon datasets show that the proposed model achieves significant improvements in the effects of the bilinear interactions in and across attributes and local text features.

1. Introduction

Regarding understanding customer feedback or reviews, the sentiment analysis technique is emerging as a viable tool for any business [1–3]. Specifically, sentiment analysis algorithms are being used to make sense of user feedback in customer reviews that contain the attitudes of the users toward products [2,4]. When a user decides whether to purchase a product, the reviews of past users can play a vital role since they may encourage more user clicks and even increase conversions [5]. Similarly, customer reviews with negative polarity will force manufacturers to improve the quality of their products.

Sentiment analysis is a fundamental task requiring a system to categorize user feedback as positive, negative, or neutral. Based on the distributed representation of words (i.e., word2vec [6,7] and GloVe [8]), previous studies recommended using neural network models, such as convolutional neural networks (CNNs) [9,10], gated recurrent units (GRUs) [11,12] and long short-term memory (LSTM) [13,14], to encode the vectors of the constituent words of the reviews for the final classification. To further enhance the performance, self-attention [15,16] and dynamic routing algorithms [17,18] can be employed to model the insightful relations between words, or a hierarchical attention network (HAN) [19] can be used to emphasize the importance of words in sentences. Pretrained language models (PLMs) [20], such as Embeddings from Language Models (ELMo) [21], Bidirectional Encoder Representations from Transformer (BERT) [22], and other variants, e.g., ALBERT [23], RoBERTa [24] and XLM-RoBERTa [25], can be fine-tuned and applied to the sentiment analysis task. These models obtain state-of-the-art results on various natural language processing (NLP) tasks, including sentiment analysis [26].

The abovementioned literatures have one thing in common in that they all use only features from the local text. Actually, there is more extra information, such as users and products, which may impact contextual, semantic, and sentiment writing [27]. When such information is involved, the sentiment classification model becomes personalized and estimates the sentiment polarity of a review that a user would write with respect to a certain product. That is, different sentiment polarities may be found in the reviews that are written by different users about the same product or by the same users about different products.

Therefore, recent studies have applied deep neural networks to incorporate external attributes, such as users, time, aspects and topics, which have been widely used and have achieved better performance in personalized or controllable text summarization [28,29] and generation [30,31]. For sentiment classification

[☆] The code for this paper is available at: https://github.com/yoyo-yun/BilinearReview_journal.

[*] Corresponding author.

E-mail addresses: yzhang@mail.ynu.edu.cn (Y. Zhang), wangjin@ynu.edu.cn (J. Wang), xjzhang@ynu.edu.cn (X. Zhang).

URLs: http://www.ise.ynu.edu.cn/teacher/973 (J. Wang), http://www.ise.ynu.edu.cn/teacher/711 (X. Zhang).

https://doi.org/10.1016/j.knosys.2021.107135
0950-7051/© 2021 Elsevier B.V. All rights reserved.

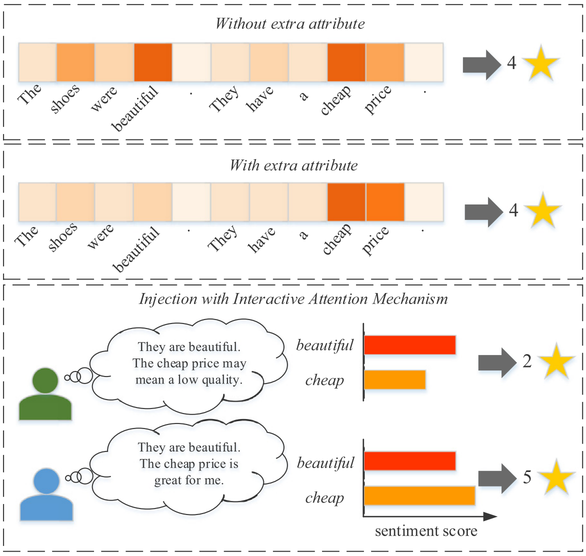

Without extra attribute
The shoes were beautiful . They have a cheap price .
4
With extra attribute
The shoes were beautiful . They have a cheap price .
4
Injection with Interactive Attention Mechanism
They are beautiful. The cheap price may mean a low quality.
beautiful
cheap
2
They are beautiful. The cheap price is great for me.
beautiful
cheap
5
sentiment score

Fig. 1. Examples of customer reviews with attributes as bias used in the sentiment classification task. A deep color represents a higher weight assigned to corresponding words according to their contribution to the final classification. Higher sentiment scores indicate more positive polarity.

tasks, user and product information can also be incorporated into neural network models [32–34]. Based on this, a series of methods were proposed following [27], which proved that the neural models can leverage such information by incorporating them into a CNN classifier to boost the performance. However, the trained preference matrix is insufficient, and the information of attribute embeddings may be ignored at the semantic level (or sentence level). Based on the HAN, Chen et al. [35] incorporated user and product information together into both word- and sentence-level attention to produce review representations for the final prediction based on a neural sentiment classification (NSC) model. Then, Wu et al. [36] observed that different words or sentences always show different biases for different users or products. They proposed a hierarchical user attention and product attention model (HUAPA) which generates different review representations from different users and products separately and then concatenates them as the final text representation.

Intuitively, the previous works only consider user and product information as separate features to be individually incorporated into the text representation. Ma et al. [37] use user and product information as bias information to reassign attribute-guide weights to words according to their contribution to the final prediction. However, these works all treated these attributes separately but ignored the interactive relationships between these attributes (such as user–product interaction). Inspired by the success of collaborative filtering in recommendation systems [38,39], simultaneously learning the representation of user and product and investigating the insightful relationship between these attributes can benefit personalized sentiment analysis. For instance, a user may have a preference for a brand of goods. Furthermore, good reviews and high ratings for high quality products with rock-bottom prices are easy for the low-income population to submit. Such implicit relations between users and products may be absent in the review context, but it can be used in sentiment classification tasks and boost performance (see Fig. 1).

In this paper, we proposed an interactive attributes attention network (IAAN) to incorporate the implicit interactive features between users and products into text representations for the sentiment classification task. In addition to the self-attention on the local text representation, three more interactive attentions are employed, including user–product, user-to-text, and product-to-text encoders, which can force the model to select more meaningful words and sentences and introduce global user and product information in an interactive form. In order to reduce the number of parameters and improve the efficiency of the training process, bilinear terms are applied instead of self-attention. To enhance the information of each attribute and the information at both the word and sentence levels, we also applied a hierarchical architecture to produce the user–text, product–text, and local text representations. In the training phase, a multiloss objective function is applied to all parts of the representations to integrate both attributes and text features into the final classification.

The empirical experiments were conducted on the IMDB, Yelp, and Amazon datasets. The comparative results indicate that the proposed model outperformed other prior methods. Another observation is that the bilinear item [40] of the proposed model outperformed other attention mechanisms, indicating its strong ability to capture interactive information of the pairs of attribute–text features.

The main contributions of this study are as follows:

- We introduce a bilinear term, which consists of the inner and Hadamard products, to effectively capture the interactive relationships between attributes and texts.
- A multiloss objective function was applied on all four different representations to achieve better performance on the integration of the representation.
- The comparative and interpretable experiments showed that our proposed model outperformed prior methods. The competitive improvements were derived from interactive attribute features.

The remaining sections of this paper are organized as follows. Section 2 briefly introduces the related works on sentiment analysis that incorporate external attributes. Section 3 formulates the overall framework of the proposed interactive attributes attention model in detail. The comparative and interpretable experiments conducted on the IMDB, Yelp, and Amazon datasets are summarized in Section 4. Finally, the conclusions are drawn in Section 5.

2. Related work

Sentiment analysis mainly aims to study human sentiment opinions oriented toward a natural language text. In this section, a brief review of the previous works on attribute incorporation and sentiment classification is presented.

2.1. Sentiment classification

Motivated by deep learning techniques [41], increasingly more neural networks are used for sentiment classification [42,43] of customer reviews and achieve competitive performances [44,45]. CNN, LSTM, and GRU are conventional models for sentence modeling. Kim [9] investigated the effect of applying a CNN to sequential datasets to capture local *n*-gram information for text classification. To consider sequence order information and capture long dependency features, LSTM and GRU are adopted to learn sentence representations, which also address the vanishing and exploding gradient problems occurring in vanilla recurrent neural networks (RNNs). These models are widely used in most structures with hierarchical architectures at both the word and sentence levels for document-level sentiment analysis.

Considering that not all words contribute equally to a final classification, a series of attentive models are used to generate structured sentence representations by highlighting the most important tokens of the sentence. For example, structured attentive LSTM [15] uses an attention matrix to map sequential information in order to locate multiple attention distributions for structured sentence representation. Instead of using the reweighting strategy of the attention mechanism, capsule networks [46] adopt a dynamic routing strategy, which considers what and how much information should be transferred from an input capsule layer to the output layer [18,47]. Specifically, Yang et al. [19] proposed a hierarchical attention network that generated document representation through an attention mechanism at both the word level and sentence level. Recently, Basiri et al. [48] introduced an Attention-based Bidirectional CNN–RNN Deep Model (ABCDM) for sentiment analysis, considering temporal information flow and different emphasis on word sequence. In addition, Wei et al. [49] used multi-polarity orthogonal attention to locate implicit sentiment information in BiLSTM.

To further improve the performance in sentiment analysis, several studies recommend ensemble learning mechanisms. For instance, Mohammadi and Shaverizade [50] and Akhtar et al. [51] proposed ensemble methods to combine representations of different models, e.g., LSTM, GRU, BiLSTM, and CNN, for sentiment analysis. Furthermore, Cambria et al. [52] introduced an ensemble neural model to learn knowledge representation from both symbolic and subsymbolic methods.

Recently, PLMs have been widely used in a variety of natural language understanding (NLU) tasks and have achieved state-of-the-art results, including in question answering, name entity recognition, text generation, and sentiment analysis. The PLMs have two phases. First, these models are trained on a large number of natural language texts in a self-supervised way (with general tasks), and then they migrate the pretrained information into downstream tasks to enhance their performance in a supervised way (with specific tasks). However, these models contain a large number of parameters, leading to huge increases in computational costs and resources along with the growth of the text length. To address these issues, a series of studies have been proposed. For example, Adhikari et al. [53] first describe the BERT model dubbed DocBERT for long-range text classification, which achieves good results using a truncation strategy. Pappagari et al. [54] proposed RoBERT and ToBERT, which used a quick fine-tuning BERT training procedure, to successfully use segmentwise predictions to conduct document classification with good improvements.

2.2. Personalized approach

External attributes, such as topics, users, products, times, and aspects, are widely used to investigate personalized information and boost the performance of neural networks on various NLP tasks [28,30,31,55,56]. For instance, Michel and Neubig [55] introduced a method to inject speaker-specific variations as an additional bias vector into the machine translation system to improve the translation quality and effect in terms of speaker traits. Dong et al. [31] proposed an attribute-to-sequence review generator given implicit attribute information, such as users, products, and ratings. Ni and McAuley [30] incorporated aspect-ware user and item representations into an aspect encoder to build a personalized review generator. Suhara et al. [29] proposed an OpinionDigest model that extracted multiple opinions from multiple reviews and selected appointed opinions for customized summarization.

Previous works have recommended that attributes play an important role in improving the performance of sentiment analysis tasks [32–34]. Tang et al. [27] first incorporated user and product information into document-level sentiment classification, achieving a great improvement by including a text preference matrix and a representation vector in CNN classification. However, this method may suffer from insufficiencies and difficulties in training in terms of the preference matrix, and the attributes can only be applied to word-level representations. Then, Chen et al. [35] proposed an NSC model with user and product attention (NSC+UPA) to address the problem. The model jointly injects external information into an attention mechanism of the hierarchical attention model where it biases the model to focus on important parts according to certain attributes on the word level and sentence level, respectively. Wu et al. [36] observed that different words or sentences may show different importances in different views (user and product). To validate the argument, they introduced a two-way structure containing two separate hierarchical attention models incorporating user and product information, respectively. Amplayo [56] proposed a more effective way to inject attributes using a flattened LSTM model instead of a hierarchical framework and investigated how and where to inject these attributes.

In summary, previous works achieved significant improvements in personalized sentiment analysis. However, all these models took the attributes as separate information incorporated into reviews and ignored the interactions between attributes and the interactions across representations from attributes and reviews. Instead, we introduce the bilinear term to perform interactions across attributes and reviews.

3. Interactive attributes attention model

In this section, we describe the proposed interactive attributes attention model in detail. Fig. 2 shows the overview of the proposed model, which mainly consists of three layers, including the embedding layer, the interactive attention layer, and the output layer. The embedding layer takes the text, user and product information as input and transforms them into distributed representations. Then, a bilinear interaction was used as the **user–product interaction** to learn the implicit relationship between users and products in the interactive layer. Both the **user-to-text encoders** and **product-to-text encoders** apply a hierarchical architecture to incorporate global user and product information in the attention at both the word and sentence levels. Similarly, the **text encoder** uses the same hierarchical architecture, except that it uses self-attention instead of the bilinear item, to capture both the text information in sentences and the sequential information between sentences. In the output layer, the four obtained representations and their concatenation were used to perform five classification tasks with a joint multiloss objective function, where the four subtasks were expected to improve the final classification performance.

3.1. Embedding layer

An input review text t of user u for product p usually contains L sentences with various text lengths. We define the maximum length of the sentences in the corpus as N. If the length of the sentence is shorter than N, it will be padded with zeros. The embedding layer transforms the sparse inputs u and p into dense real-valued representations, $e^u \in \mathbb{R}^{d_u}$ and $e^p \in \mathbb{R}^{d_p}$, where d_u and d_p respectively denote the dimensionality of the user and product vectors. It is worth noting that both e^u and e^p are randomly initialized and jointly updated during the training process. For the lth sentence, this layer can also obtain its pretraining word representation (e.g., word2vec, GloVe, ELMo or BERT) $\mathbf{x}^l = [x_1^l, x_2^l, \ldots, x_n^l, \ldots, x_N^l]$, where $x_n^l \in \mathbb{R}^{d_w}$, and d_w is the dimensionality of word embeddings.

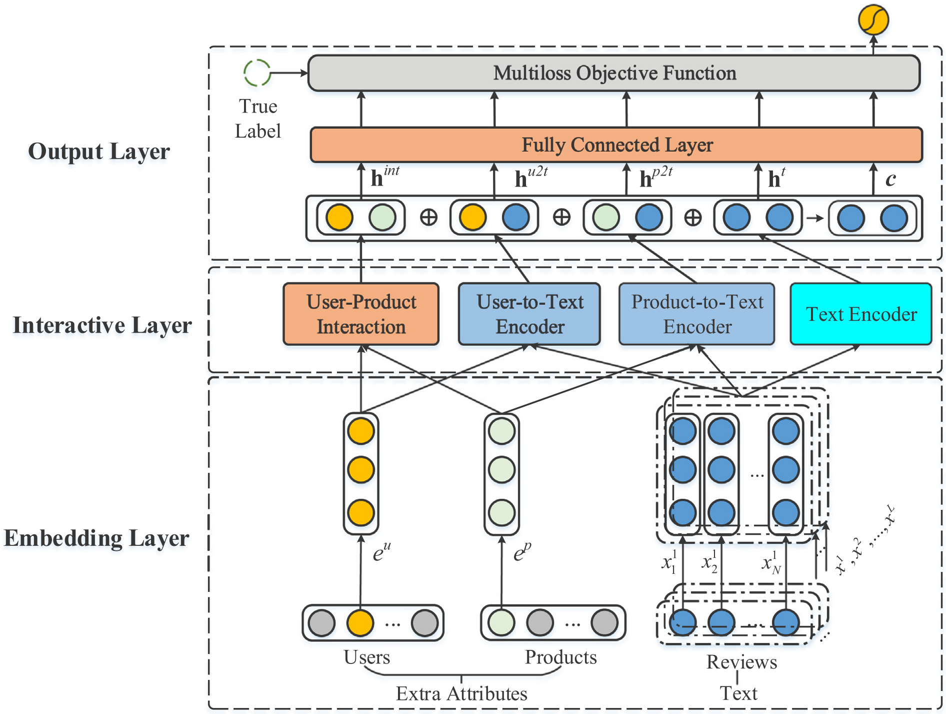


Fig. 2. Conceptual illustration of the interactive attention model for the sentiment classification of customer reviews.

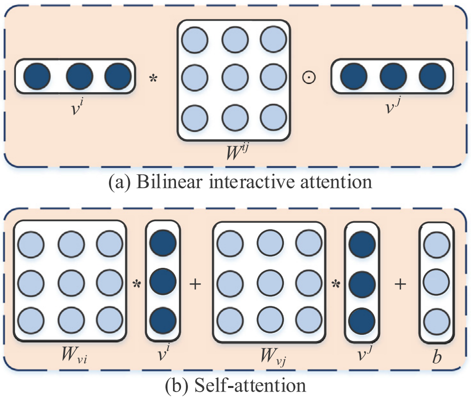


Fig. 3. Conceptual illustration of the interactive attention model for the sentiment classification of customer.

3.2. Interactive layer

Inspired by the success of feature interaction in recommendation systems, we explore the incorporation of user and product information into text representation in the interactive layer. The details of each component are described as follows.

3.2.1. User–product interaction

The user–product interaction applied a bilinear term with the inner and Hadamard product operations to learn the inner features between users and products, as shown in Fig. 3(a). Inspired by the success of collaborative filtering in recommendation systems, learning user–product information can model the preferences of different users on different products [38]. Further personalized prediction for sentiment analysis can benefit from the interactions between user and product. Additionally, such an operation needs fewer trainable parameters but can provide more efficient training and better interactive performance than self-attention, as shown in Fig. 3(b). Since self-attention takes user or product as a bias item. It needs two matrices of parameters to project the input two representations into a shared latent variable. In contrast, bilinear operation takes only one parameter matrix with multiplication. Thus, the computational efficiency of bilinear interaction is better than that of self-attention, which has been reported in the previous work [40].

By inputting user embeddings e^u and product embeddings e^p, the bilinear interaction produces the representations $\mathbf{h}^{u2p} \in \mathbb{R}^{d_p}$ and $\mathbf{h}^{p2u} \in \mathbb{R}^{d_u}$ between users and products, which are respectively denoted as

$$\mathbf{h}^{u2p} = e^u * W^{u2p} \odot e^p \tag{1}$$

$$\mathbf{h}^{p2u} = e^p * W^{p2u} \odot e^u \tag{2}$$

where $W^{u2p} \in \mathbb{R}^{d_u \times d_p}$ and $W^{p2u} \in \mathbb{R}^{d_p \times d_u}$ are the weight matrices and * and $\odot$ represent the inner and Hadamard product operations, respectively. For the different combinations of these two attributes in order, there are different parameters of the weight matrix in a bilinear term, i.e., W^{u2p} and W^{p2u}. The output of the user–product interaction $\mathbf{h}^{int}$ is a concatenation of both $\mathbf{h}^{u2p}$ and $\mathbf{h}^{p2u}$, denoted as

$$\mathbf{h}^{int} = [\mathbf{h}^{u2p}; \mathbf{h}^{p2u}] \tag{3}$$

where [;] represents the concatenate operation.

3.2.2. User-to-text and product-to-text encoder

It is obvious that not all words and sentences contribute equally to the text meaning for different users and products. The user or product embeddings were applied to guide the text representation by using either user-to-text or product-to-text encoders. The architecture of both encoders is shown in Fig. 4.

For the user-to-text encoder, a bidirectional LSTM (BiLSTM) was first applied to generate the hidden representation for the nth word in the lth sentence on the word level, which is defined as

$$\overrightarrow{\mathbf{q}}^{\,l}_{n} = \overrightarrow{LSTM}(\overrightarrow{\mathbf{q}}^{\,l}_{n-1}, x^l_n), \quad \overleftarrow{\mathbf{q}}^{\,l}_{n} = \overleftarrow{LSTM}(\overleftarrow{\mathbf{q}}^{\,l}_{n+1}, x^l_n) \tag{4}$$

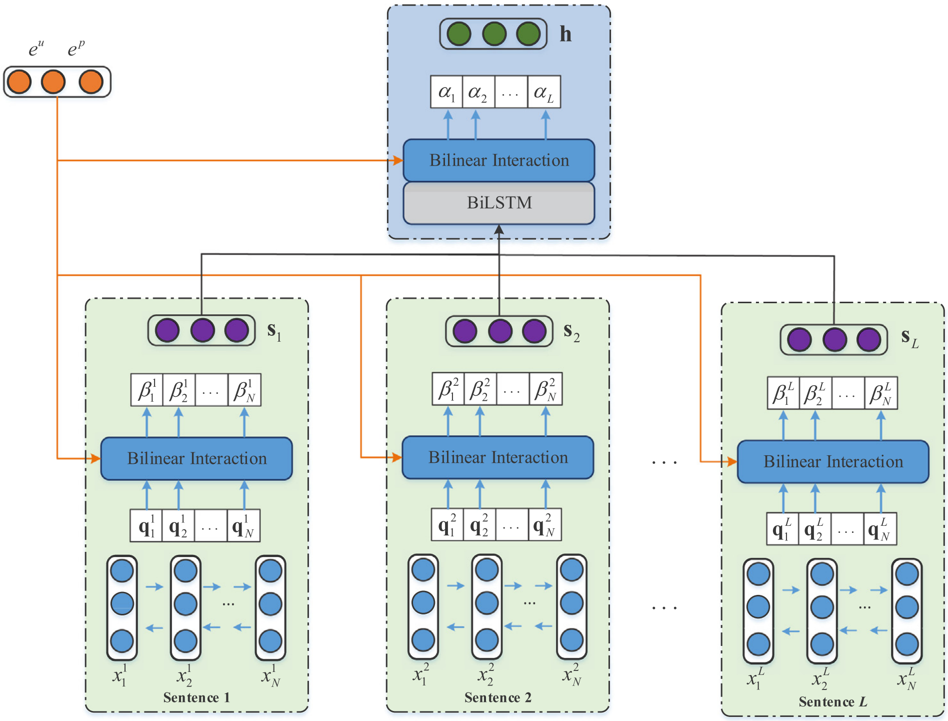

e^u e^p
$\mathbf{h}$
α_1 α_2 ⋯ α_L
Bilinear Interaction
BiLSTM
$\mathbf{s}_1$
β_1^1 β_2^1 ⋯ β_N^1
Bilinear Interaction
$\mathbf{q}_1^1$ $\mathbf{q}_2^1$ ⋯ $\mathbf{q}_N^1$
x_1^1 x_2^1 x_N^1
Sentence 1
$\mathbf{s}_2$
β_1^2 β_2^2 ⋯ β_N^2
Bilinear Interaction
$\mathbf{q}_1^2$ $\mathbf{q}_2^2$ ⋯ $\mathbf{q}_N^2$
x_1^2 x_2^2 x_N^2
Sentence 2
$\mathbf{s}_L$
β_1^L β_2^L ⋯ β_N^L
Bilinear Interaction
$\mathbf{q}_1^L$ $\mathbf{q}_2^L$ ⋯ $\mathbf{q}_N^L$
x_1^L x_2^L x_N^L
Sentence L

Fig. 4. Conceptual illustration of interactive layer for the user-to-text or product-to-text encoder.

$$\mathbf{q}_n^l = [\overrightarrow{\mathbf{q}}_n^l; \overleftarrow{\mathbf{q}}_n^l] \tag{5}$$

where $\overrightarrow{\mathbf{q}}_n^l \in \mathbb{R}^{d_q}$ and $\overleftarrow{\mathbf{q}}_n^l \in \mathbb{R}^{d_q}$ respectively denote the forward and backward representations, which are calculated separately, and d_q is the dimensionality of the hidden representation.

The user embeddings e^u are then applied to each constituent word of the lth sentence by using the aforementioned bilinear operation, denoted as

$$\mathbf{r}_n^l = e^u * W^{u2q} \odot \mathbf{q}_n^l \tag{6}$$

where $\mathbf{r}_n^l \in \mathbb{R}^{d_q}$ is the interactive representation; and $W^{u2q} \in \mathbb{R}^{d_u \times d_q}$ is the weight matrix, which is shared by all words in all sentences. The obtained interactive representation is then inputted into a *softmax* normalization function for each hidden state to extract the words that were relevant to the user information,

$$\beta_n^l = \frac{\exp\left[V^{\beta\top} \tanh(W^{\beta}\mathbf{r}_n^l + b^{\beta})\right]}{\sum_{i=1}^{N} \exp\left[V^{\beta\top} \tanh(W^{\beta}\mathbf{r}_i^l + b^{\beta})\right]} \tag{7}$$

where V^{β} and W^{β} are respectively the weight vector and matrix, and b^{β} is the bias. The resulting user-to-text representation of the lth sentence can be denoted as

$$\mathbf{s}_l^{u2t} = \sum_{n=1}^{N} \beta_n^l(\mathbf{r}_n^l + \mathbf{q}_n^l) \tag{8}$$

To reward sentences that correctly locate the most relevant information, we applied another similar bilinear interaction on the sentence level to inject the user information and measure the importance of the sentence, defined as

$$[\mathbf{g}_1^{u2t}, \mathbf{g}_2^{u2t}, \ldots, \mathbf{g}_L^{u2t}] = BiLSTM([\mathbf{s}_1^{u2t}, \mathbf{s}_2^{u2t}, \ldots, \mathbf{s}_L^{u2t}]) \tag{9}$$

$$\mathbf{f}_l^{u2t} = e^u * W^{u2t} \odot \mathbf{g}_l^{u2t} \tag{10}$$

$$\alpha_l = \frac{\exp\left[V^{\alpha\top} \tanh(W^{\alpha}\mathbf{f}_l^{u2t} + b^{\alpha})\right]}{\sum_{k=1}^{L} \exp\left[V^{\alpha\top} \tanh(W^{\alpha}\mathbf{f}_k^{u2t} + b^{\alpha})\right]} \tag{11}$$

where $\mathbf{g}_l^{u2t} \in {}^{d_g}$ is the hidden state of the lth sentence. W^{u2t} is the trainable matrix of bilinear terms, which is shared by all hidden states. W^{α}, V^{α} and b^{α} are the weight vector, matrix, and bias, respectively, which are associated with the bilinear interaction, and α_l is the attention weight of the lth sentence representation. The final user-to-text representation $\mathbf{h}^{u2t} \in \mathbb{R}^{d_g}$ can be obtained by

$$\mathbf{h}^{u2t} = \sum_{l=1}^{L} \alpha_l(\mathbf{f}_l^{u2t} + \mathbf{g}_l^{u2t}) \tag{12}$$

Similar to the user-to-text encoder, a product-to-text encoder also uses the same hierarchical architecture. It takes the product embedding e^p and the embeddings of constituent words $\left[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L\right]$ as input and applies bilinear terms to inject the product information into the text representation at both the word and sentence levels. The obtained product-to-text representation can be defined as $\mathbf{h}^{p2t}$.

3.2.3. Text encoder

The text encoder, which is considered a hierarchical attention network, is used to generate local text representations from only text features [19]. This module is somewhat similar to the user-to-text and product-to-text encoders except for replacing the bilinear interaction with self-attention and isolating the local texts from the user and product information.

Given the hidden states $\mathbf{q}_n^l$ of the nth word in the lth sentence on the word level, which is generated by the BiLSTM in Eq. (4) and (5), self-attention is introduced to extract words that are important to the semantic meaning of the sentence and aggregate the representation of these words into a sentence vector.

$$att_n^l = \tanh(W^{\gamma}\mathbf{q}_n^l + b^{\gamma}) \tag{13}$$

Table 1
Statistics of the IMDB, Yelp-2013 and Amazon datasets in three subaspects.

Datasets	#classes	#docs	#users	#products	#docs/user	#docs/product	#sents/doc	#words/sent
IMDB	10	84,919	1,310	1,635	64.82	51.94	16.08	24.54
Yelp-2014	5	231,163	4,818	4,194	47.97	55.11	11.41	17.26
Digital Music	5	169,303	16,561	11,797	8.75	12.28	2.65	12.21
Industrial	5	75,620	11,040	5,336	6.52	13.49	3.38	12.97
Software	5	12,783	1826	802	6.49	14.76	10.10	17.22

$$\gamma_n^l = \frac{\exp(V^{\gamma\top} att_n^l)}{\sum_{i=1}^{N} \exp(V^{\gamma\top} att_i^l)} \tag{14}$$

$$\mathbf{s}_l^t = \sum_{n=1}^{N} \gamma_n^l \mathbf{q}_n^l \tag{15}$$

where V^γ, W^γ and b^γ are trainable parameters; and att_n^l is the attention score for the nth words, which is then normalized by a *softmax* function. Then, the sentence representation $\mathbf{s}_l^t$ is calculated as a weighted sum of the hidden states of words according to the attention scores.

Then, the local text representation can be obtained similarly. Another BiLSTM with self-attention was also used to encode the sentences.

$$[\mathbf{g}_1^t, \mathbf{g}_2^t, \ldots, \mathbf{g}_l^t] = BiLSTM([\mathbf{s}_1^t, \mathbf{s}_2^t, \ldots, \mathbf{s}_l^t]) \tag{16}$$

$$att_l^t = \tanh(W^\varphi \mathbf{s}_l^t + b^\varphi) \tag{17}$$

$$\varphi_l = \frac{\exp(V^{\varphi\top} att_l^t)}{\sum_{k=1}^{L} \exp(V^{\varphi\top} att_k^t)} \tag{18}$$

$$\mathbf{h}^t = \sum_{l=1}^{N} \varphi_l \mathbf{g}_l^t \tag{19}$$

where V^φ, W^φ and b^φ are trainable parameters; and $\mathbf{h}^t$ is the obtained local text representation that summarizes all the information of sentences in a text.

3.3. Multiloss objective function

For training, we designed a main task to output the final result, and we also designed four subtasks to enhance the performance of the main task via multitask learning.

For the main task, we concatenate all representations of user–product interactions, the user-to-text encoder, the product-to-text encoder, and the text encoder (i.e., $\mathbf{h}^{\text{int}}$, $\mathbf{h}^{u2t}$, $\mathbf{h}^{p2t}$, and $\mathbf{h}^t$) to obtain a combination vector $\mathbf{c}$ for the final classification, formulated as follows:

$$\mathbf{c} = [\mathbf{h}^{\text{int}}; \mathbf{h}^{u2t}; \mathbf{h}^{p2t}; \mathbf{h}^t] \tag{20}$$

Then, the classification is a one-layer MLP network with a *softmax* function. The loss of the main task is a categorical cross-entropy, defined as

$$y_d^c = softmax(\text{MLP}(\mathbf{c})) \tag{21}$$

$$\mathcal{L}^c = -\sum_{d=1}^{D} \mathbb{I}(y_d) \log(y_d^c) \tag{22}$$

where y_d and y_d^c represent the ground truth and the probability distribution of sample d, respectively. D is the number of training samples, $\mathbb{I}(y)$ denotes a one-hot vector with the yth component being one.

To further enhance the performance, we applied a multiloss strategy that separately input representations (i.e., $\mathbf{h}^{\text{int}}$, $\mathbf{h}^{u2t}$, $\mathbf{h}^{p2t}$, and $\mathbf{h}^t$) for the same training objective. Taking user–product representations as an example, the loss function is a categorical cross-entropy, defined as

$$\mathcal{L}^{int} = -\sum_{d=1}^{D} \mathbb{I}(y_d) \log(softmax(\text{MLP}(\mathbf{h}^{int}))) \tag{23}$$

Similarly, the other three subtasks can obtain three more losses, defined as $\mathcal{L}^{u2t}$, $\mathcal{L}^{p2t}$ and $\mathcal{L}^t$. The corresponding final objective of subtasks is a weighted sum as follows:

$$\mathcal{L}^{sub} = \lambda_{int}\mathcal{L}^{int} + \lambda_{u2t}\mathcal{L}^{u2t} + \lambda_{p2t}\mathcal{L}^{p2t} + \lambda_t\mathcal{L}^t \tag{24}$$

where λ_{int}, λ_{u2t}, λ_{p2t}, and λ_t are the coefficients for each respective subtask. The final loss is a weighted sum of $\mathcal{L}^c$ and $\mathcal{L}^{sub}$, defined as

$$\mathcal{L} = \lambda_c\mathcal{L}^c + \mathcal{L}^{sub} \tag{25}$$

where λ_c is the corresponding loss coefficient of the main task.

4. Comparative experiments

The comparative experiments were conducted on customer review datasets with extra attributes to evaluate the performance of the proposed personalized sentiment classification model against the previous state-of-the-art models.

4.1. Datasets and evaluation metrics

Following Tang et al. [27], the IMDB, Yelp-2014, and Amazon datasets with user and product information were used for the experiments. For the Amazon datasets (2018 version[1]), three subsets of data were extracted, including Digital Music, Industrial and Scientific, and Software. All datasets were randomly divided into training, dev and test sets at a ratio of 8:1:1. The detailed statistics of the datasets are summarized in Table 1.

Two evaluation metrics were employed to measure the final classification performance and divergences between the predicted and ground-truth labels.

- Accuracy (*Acc*):

$$Acc = \frac{\sum_{i=1}^{N} P_i}{N} \tag{26}$$

- Root mean squared error (*RMSE*):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{27}$$

where N is the total number of examples, and P_i has a value of 0 or 1. When the prediction of example i belongs to the ground truth class, P_i equals 1; otherwise, P_i equals 0. $\hat{y}_i$ and y_i denote the predicted and ground truth labels on example i, respectively. Note that lower *RMSE* scores mean closer distribution between predicted and ground truth labels. Higher *Acc* and lower *RMSE* scores represent better performance.

[1] https://nijianmo.github.io/amazon/index.html.

4.2. Baselines

To evaluate the performance of the proposed model, a series of previous works were selected as baselines for comparison. All baselines were categorized according to whether extra attributes were used or not. The baseline information is summarized as follows.

(1) Baselines without attribute information:

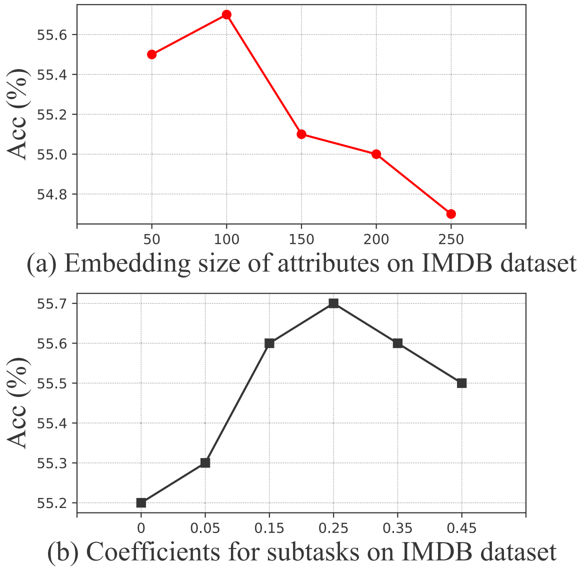
- The **CNN** [9] uses a simple CNN classifier to determine the sentiment polarity for a given document. It is also the same model as the **UPNN (CNN w/o UP)**, which will be introduced later.
- **BiLSTM** [57]. It uses **LSTM** [14] to model both past and forward directions to learn document representations for sentiment analysis. Both the CNN and BiLSTM are used as a base model and more layers are stacked on them to obtain a structured architecture.
- **NSC+LA** [35]. The NSC model with local text features uses LSTM as a base model to structure a hierarchical attention model at the word and sentence levels.
- **DocBERT** [53]. It refine-tunes the pretrained BERT for document-level sentiment analysis in a simple way by truncating the original long text into text with a limited maximum length (512 tokens).

(2) Baselines with attribute information:

- **JMARS** [58]. The joint model with aspects, ratings and sentiment uses user information and aspect information to conduct collaborative filtering and topic modeling for sentiment analysis.
- **UPNN (CNN)** [27]. The user product neural network takes a CNN classifier as the base model and incorporates the attribute information for final predictions by introducing a word-level preference matrix into word embeddings and an attribute representation vector into the final classifier. By extension, the **UPNN (NSC)** is constructed using the NSC architecture and adopts the same method of incorporating attributes as the **UPNN (CNN)**.
- **NSC+UPA (BiLSTM)**. The NSC model with user and product information uses a hierarchical BiLSTM model with the user and product-specific attention mechanism for document-level sentiment analysis by jointly incorporating attribute information into the attention mechanism to locate attribute-specific focuses on reviews.
- **HUAPA** [36]. The hierarchical model with user attention and product attention incorporates user and product information into the attention mechanism of the hierarchical attention model separately and concatenates different attribute-specific review representations into a final classifier to predict sentiment ratings.
- **CHIM** [56]. The chunkwise importance matrix model introduces a weight matrix of user and product information to investigate the effect of injecting attributes at different places (embedding, encoder, attention, and classifier) to achieve improvements.
- **IAAN**. The proposed interactive attribute attention network was also implemented for comparison. By using bilinear interaction, the model can be used to extract the relationships between users and products and incorporate the information into the final representation for classification.

4.3. Implementation details

The embedding of each attribute (e^u or e^p) was initialized from a uniform distribution $U(-0.01, 0.01)$ with a dimension size of 100 and updated in the training phase. Regarding the word vectors, GloVe[2] [8] pretrained on the 840B Common Crawl corpus with a dimension size (d_w) of 300 was applied. Regarding the hierarchical framework, the hidden size of BiLSTM (d_q, d_g) was set to 200 at either the word or sentence level. The Adam optimizer with a base learning rate of 5e−4 was used to update all trainable parameters. We also employed an early stopping strategy [59] with a patience of 3 to avoid overfitting. The coefficient weight of each subtask classifier loss was equally set to 0.25 and that of the main classifier was set to 0.35. Note that dropout [60] with a rate of 0.2 was taken as a regularization strategy to achieve better performance.



(a) Embedding size of attributes on IMDB dataset

(b) Coefficients for subtasks on IMDB dataset

Fig. 5. *Acc* scores of the model for the IMDB dev split with respect to various attribute dimensions and different attribute dimensions.

To make a fair comparison between the proposed models and other baselines and extend them to the Amazon datasets, we implemented the previous state-of-the-art models and reported the mean results (marked with a star *) of five runs. These previous models use a similar method of incorporating attributes as the proposed models.

4.4. Hyperparameter fine-tuning

As bilinear interaction was introduced in user–product and attribute–text representations, the dimension of each attribute (d_u or d_p) was shown to be a sentiment factor for customized review information. Therefore, the optimal attribute dimension that achieved the ultimate performance was investigated using different dimension settings ranging from 50 to 250 for to IMDB dev split. Note that the user and product embedding sizes were set equal. The classification *Acc* influenced by different attribute dimensions is shown in Fig. 5(a). As indicated, the performance of the proposed model increases as the attribute dimension starts to increase but declines when the attribute embedding size exceeds 100. This phenomenon demonstrates that an appropriate attribute embedding size is important to improve personalized review classification. A smaller dimensionality may capture weak information and make it difficult to leverage the attribute characteristics; however, a larger dimensionality will produce more biased information in the local text representation, resulting in decreased performance.

[2] http://github.com/stanfordnlp/GloVe.

Table 2
Comparative results of the proposed model and the baselines. The **boldfaced** values represent the best results in different parts. Results marked with stars (–*) correspond to reimplementation performance.

Models		IMDB		Yelp-2014		Digital Music		Industrial		Software	
		Acc	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*
Without attributes	LSTM	37.8	1.612	56.3	0.769	83.5	0.585	76.7	0.794	60.1	1.127
	BiLSTM	43.3	1.494	59.2	0.733	**84.0**	**0.569**	77.2	0.769	60.3	1.060
	UPNN (CNN w/o UP)	40.5	1.629	58.5	0.808	83.5*	0.577*	77.2*	0.728*	65.4*	1.018*
	NSC	44.3	1.465	63.7	0.686	–	–	–	–	–	–
	NSC+LA	**48.7**	**1.381**	63.0	0.715	**84.0***	0.582*	**77.6***	**0.711***	65.7*	0.982*
	DocBERT	47.4	1.399	**67.0**	**0.625**	80.0	0.583	76.4	0.801	**68.5**	**0.901**
With attributes	JMARS	–	1.773	–	0.999	–	–	–	–	–	–
	UPNN (CNN w/ UP)	43.5	1.602	60.8	0.764	85.7*	0.594*	76.8*	0.944*	61.2*	1.240*
	UPNN (NSC)	47.1	1.443	–	–	–	–	–	–	–	–
	NSC+UPA (BiLSTM)	53.3	1.281	66.7	0.654	85.5*	0.521*	78.8*	0.680*	66.0*	1.014*
	HUAPA	55.0	1.185	68.6	0.626	85.4*	0.544*	79.1*	0.670*	69.9*	0.859*
	CHIM$_{embedding}$	**56.4**	1.161	69.2	0.629	–	–	–	–	–	–
	IAAN	**56.4**	**1.158**	**69.4**	**0.621**	**87.3**	**0.477**	**81.1**	**0.669**	**70.5**	**0.877**

To investigate the effect of different weights of subtask losses on the final performance of the concatenation representation, experiments were conducted on various loss coefficients, and the results are illustrated in Fig. 5(b). The base learning rate was set to 5e−4. The coefficient for the loss of the main task (λ_c) was fixed to 0.35, and all coefficients for the subtask losses ($\mathcal{L}^{u2t}$, $\mathcal{L}^{p2t}$ and $\mathcal{L}^{t}$) were set equally and fine-tuned from 0 to 0.45. Once $\mathcal{L}^{u2t}$, $\mathcal{L}^{p2t}$ and $\mathcal{L}^{t}$ were set to 0 in the model, the model was trained only by the loss of the main task, which achieved the worst *Acc* score. This suggests that the losses of the subtasks can improve the final performance. Considering various objective functions as different tasks, the improvements can be successfully located in the multitask learning method, which improves the model's generalizability to achieve better classification [61]. As the coefficient of the subtasks increased, the performance of the proposed model improved. Once the coefficient exceeded 0.25, the *Acc* declined. However, it is still higher than the initial state with $\mathcal{L}^{u2t}$, $\mathcal{L}^{p2t}$ and $\mathcal{L}^{t}$ as 0. The results show that introducing subtask losses is beneficial to enhancing the classification performance via multitask learning. In addition, appropriate weights that were applied to the multiloss objective function can greatly improve the performance.

4.5. Comparative results

Table 2 shows the comparative results against several previously proposed baselines, which were categorized according to whether attribute information was used.

For baselines without attribute information, some conventional neural models, such as the CNN and (Bi)LSTM models, have a strong ability to encode texts. However, NSC outperformed (Bi)LSTM mainly because the hierarchical structures can boost the performance. By further introducing an attention mechanism to integrate sequential information at both the word and sentence levels, NSC+LA outperformed NSC, indicating that structured information is beneficial for classification. Such a hierarchical attention structure is also the prototype of the proposed model and most of the previous state-of-the-art models. The DocBERT model was also applied to all three datasets for comparison. As shown, DocBERT achieved the best results on the Yelp-2014 and Software datasets. The improvement is located in the pretrained model migrating ample knowledge from a large corpus to downstream tasks.

After the attribute information is incorporated, the classification performance of the baselines becomes better than that of the models without attribute information. For example, the UPNN achieved a better result than its variation without user and product information. Moreover, the NSC+UPA model outperformed its basic model, i.e., NSC+LA, due to the successful introduction of extra information into the attention mechanism to guide attribute-specific attention biases on the hierarchical architecture. Furthermore, HUAPA outperformed NSC+UPA, indicating that the hierarchical model incorporating user and product information separately can generate reasonable text representations.

The proposed IAAN achieved the best performance on all datasets. Compared with the baselines without extra information, the IAAN can leverage implicit attribute features to boost the performance of the personalized sentiment classification of customer reviews. Due to the large knowledge migration from pretraining, DocBERT outperformed conventional neural models e.g., CNN, LSTM and BiLSTM on most datasets. Unfortunately, implicit information such as user preferences and product characteristics cannot be simply incorporated; therefore, the performance of DocBERT is still lower than that of the proposed IAAN model. Furthermore, the IAAN achieved better results than the UPNN (CNN), NSC+UPA and HUAPA. The main reasons include the following: (1) Hierarchical attention models, which are proven to be effective structures to encode texts, are adopted as the base models. (2) Based on HUAPA, incorporating user and product information separately was also beneficial to the proposed model, where the sentiment part is interactive attribute attention with bilinear interaction replacing the original attentive strategy. (3) Introducing user–product interaction into the final representation can boost the performance.

4.6. The effects of different attentions

To analyze the effect of different attentions, we compare bilinear interaction with the standard one dubbed concatenate attention, which was widely used in the previous works, as shown in Fig. 3(b). Table 3 shows the comparative results of using different attentions to incorporate user or product information based on the NSC structure, which represents the hierarchical structure [35] and is also the basic model of NSC+UPA, HUAPA and the proposed IAAN models.

As indicated, incorporating attribute information, user information and product information can improve the classification performance without any attribute information (N/A). Compared with concatenation attention-based models (CON), models with bilinear attention (BIL) achieved better results, indicating that the bilinear interaction facilitates the representation of reviews more effectively than the previous methods. Furthermore, the strategy of applying bilinear attention in the user–text and concatenation in the product–text bias achieved the best results. The possible reason is that customer reviews always contain sentimental representations of the personalization of a specific user. As a result, injecting user information into review representations through

Table 3
Comparative results based on different attention mechanisms. N/A means no attributes incorporation, and CON and BIL represent the concatenation and bilinear attention mechanism for user and product attributes, respectively.

Attribute		IDMB		Yelp-2014		Digital Music		Industrial		Software	
User	Product	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc.*	*RMSE*	*Acc*	*RMSE*
N/A	N/A	48.7	1.332	64.7	0.672	84.0	0.582	77.6	0.711	65.7	0.982
CON	CON	54.1	1.239	67.9	**0.638**	85.2	0.531	78.3	0.711	67.9	0.952
BIL	BIL	54.3	1.231	67.9	**0.638**	86.4	0.516	79.7	0.729	68.5	0.942
BIL	CON	**55.2**	**1.210**	**68.3**	0.643	**87.2**	**0.488**	**80.5**	**0.704**	**68.7**	**0.898**
CON	BIL	53.5	1.220	67.1	0.665	86.0	0.504	78.7	0.725	66.9	0.981

Table 4
Comparative results of different word embeddings.

Models	IDMB		Yelp-2014		Digital Music		Industrial		Software	
	Acc	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*
IAAN$_{GloVe}$ (dim = 50)	53.9	1.299	65.6	0.666	86.4	0.511	78.9	0.754	66.5	0.996
IAAN$_{GloVe}$ (dim = 100)	55.8	1.187	66.9	0.649	86.6	0.507	79.6	0.701	67.1	0.980
IAAN$_{GloVe}$ (dim = 200)	55.8	1.172	67.7	0.646	87.0	0.496	80.3	0.696	69.3	0.919
IAAN$_{GloVe}$ (dim = 300)	56.4	1.158	69.4	0.621	87.3	0.477	81.1	0.669	70.5	0.877
IAAN$_{Word2vec}$ (dim=200)	55.5	1.183	68.7	0.627	–	–	–	–	–	–
IAAN$_{BERT}$ (dim = 768)	57.5	1.087	70.1	0.605	87.8	0.468	82.2	0.592	71.8	0.852

bilinear interactive attention is more effective than concatenation. In contrast, incorporating product features as bias is more sensible in a concatenation attention mechanism.

4.7. The effects of different word representations

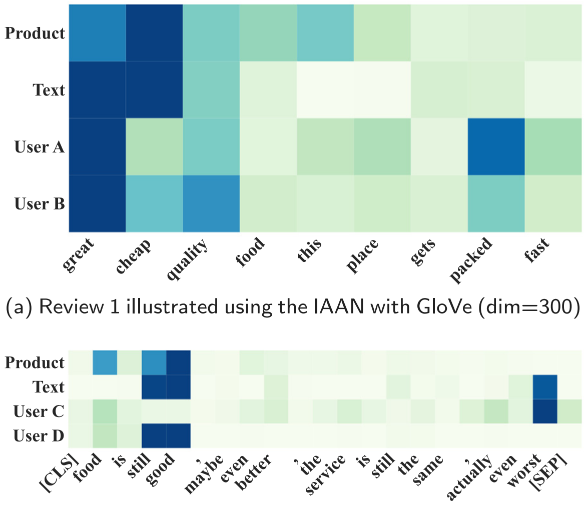
For specific tasks, different word embedding strategies, ranging from word2vec to BERT, can lead to different representations. Therefore, we investigated whether different word representations may impact the final performance. Therefore, following the previous work [36], the 200-dimensional word2vec is pre-trained on each dataset (only IMDB and Yelp-2014) with the skip-gram [7]. GloVe with different embedding sizes is introduced using off-the-shelf open sources.[3] For BERT, we use the features from the uncased base model. The comparative results were listed in Table 4.

For GloVe, as the embedding dimensionality increases, the proposed model achieves a slight improvement in classification performance on all three kinds of datasets. This suggests that larger embedding sizes leverage more information in word representation for performance gains. Different embeddings with the same dimension, i.e., GloVe and Word2vec, achieve very similar performances. However, both GloVe and Word2vec are noncontextualized word representations, resulting in a certain limitation in capturing more meaningful information. In contrast, PLMs, i.e., BERT, are contextualized models that achieve the best performance since they can obtain contextualized word embeddings for more improvements.

4.8. Ablation experiments

In order to analyze the effect of the main tasks and the corresponding four subtasks of the proposed models, ablation experiments were conducted. The results of the ablation experiments are shown in Table 5. For the implementation, we successively removed one of the subtasks, i.e., user–product, user–text, product–text, and local text representations, to investigate whether it degraded the performance of the proposed IAAN model.

As indicated, the ablation models experienced different degrees of decreased performance, indicating that each subtask of the IAAN model plays an indispensable role in performance improvements. Removing either user–text or product–text representations leads to a certain performance decrease, indicating that both user and product attributes can help the model understand personality information and product characters. In addition, introducing user–product interaction representations can be beneficial for the final classification, indicating that the implicit information between users and products could interact to improve the sentiment classification performance of customer reviews.



(a) Review 1 illustrated using the IAAN with GloVe (dim=300)

(b) Review 2 illustrated using the IAAN with BERT-based word representation.

Fig. 6. Visualization of attention over words in local text, products, and different users (denoted as User b A, User B, User C and User D).

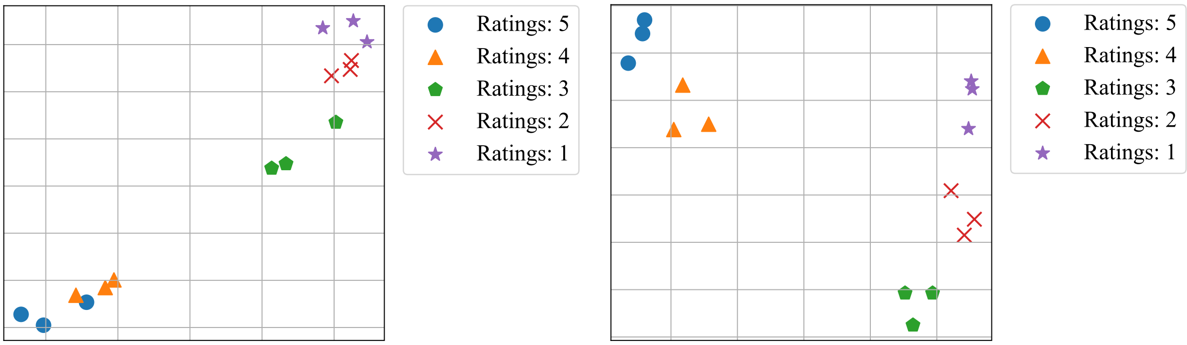
4.9. Case studies (visualization) and discussion

To intuitively assess the effectiveness of the proposed models in capturing sentiment information, several case studies are presented. In detail, two samples are randomly selected from the test split of the Yelp-2014 dataset for attention visualization. A deeper color indicated a higher attention score. For further analysis, illustrations were given in different word embeddings with respect to various users in Fig. 6(a) and Fig. 6(b).

[3] https://github.com/huggingface/transformers.

Table 5
Ablation experiment results for component analysis.

Models	IMDB		Yelp-2014		Digital Music		Industrial		Software	
	Acc	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*	*Acc*	*RMSE*
IAAN	56.4	1.158	69.4	0.621	87.3	0.477	80.0	0.651	70.5	0.877
IAAN w/o user–product	55.3	1.203	68.2	0.638	87.0	0.491	78.8	0.729	69.4	0.859
IAAN w/o user–text	54.5	1.172	68.1	0.642	85.7	0.502	77.1	0.721	65.0	0.975
IAAN w/o product–text	55.0	1.192	68.1	0.637	87.1	0.497	79.4	0.680	68.8	0.891
IAAN w/o local text	55.2	1.181	67.9	0.631	86.9	0.497	79.7	0.690	69.2	0.895



(a) Users w/ a certain product
(Product id: ryvMJK6AlbU35HKrlFT61w).

(b) Products w/ a certain user
(User id: kSVYpNWA19wUplbdi0U0Uw).

Fig. 7. The t-SNE Visualization of user and product embeddings derived from the proposed model. Each point represents a user (or a product) from Yelp-2014 datasets. The points with the same color denote the users (or products) with the same ratings for the certain product (or the user).

In Review 1, both **User A** and **User B** pay attention to *great* for *food*. Meanwhile, **User A** tends to care about the *packedfast* service while **User B** cares about the *cheap* price, resulting in 4 stars from **User A** and 3 stars from **User B**. Similarly, in Review 2, **User D** gives a rating of 3 stars for *goodfood* while **User C** only submits 1 star due to *bad* service. Additionally, the local text representations in both reviews focus their attention on both descriptions of *food* and *package*. With product interaction, *food* is then the main item of the product, resulting in higher attention scores by incorporating product information. In addition, different users may focus on different perspectives from the original attention distributions and then align with the corresponding preferences, resulting in different attention distributions in the user–text interactive representation.

To investigate how user and product embeddings learned from user–product interaction can facilitate the final classification, we conducted a visualization experiment to show the relationship between users and products, as shown in Fig. 7. We randomly selected a certain product (or user) and the representations of its various relevant users (or products) with the ratings from Yelp-2014 datasets. The t-SNE algorithm was used as a visualization tool to reduce the dimensionality of these representations. As indicated, all user or product embeddings with close ratings were located close to each other in the feature space. Moreover, the larger difference the ratings were, the farther the user (or product) embeddings were in the space for a certain product (or user). Notably, these representations of attributes were randomly initialized and then simultaneously updated in the training process of the proposed model. As similar to the collaborative filtering in recommendation systems, both representations can model the preference of different users toward different products. Therefore, further personalized prediction for sentiment analysis can benefit from the interactions between user and product.

5. Conclusion

In this paper, an interactive attribute attention model was proposed for personalized sentiment analysis. It applied attribute interactions across attributes and reviews instead of treating them separately. By using a bilinear term, which consists of the inner and Hadamard products, the model can effectively capture the interactive relationships between the two input representations. To achieve better representation integration performance, a multiloss objective function was applied, which trained four subtasks to improve the sentiment classification performance. Experimental results on the IMDB, Yelp, and Amazon demonstrate the effectiveness of the proposed models compared to the previous methods.

In addition to user and product information, other attributes can also be injected in the same way. In the future, we will attempt to investigate the performance of the proposed model in a wild field of NLPs and to migrate the interaction between implicit features into pretrained models, such as the BERT and GPT, for various natural language understanding tasks.

CRediT authorship contribution statement

You Zhang: Investigation, Methodology, Software, Formal analysis, Validation, Writing - original draft. **Jin Wang:** Conceptualization, Software, Formal analysis, Resource, Writing - review & editing, Resources, Funding acquisition. **Xuejie Zhang:** Project administration, Resources, 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 work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61702443, 61966038 and 61762091. The authors would like to thank the anonymous reviewers for their constructive comments.

References

[1] L.M. Rojas-Barahona, Deep learning for sentiment analysis, Lang. Linguist. Compass 10 (12) (2016) 701–719, http://dx.doi.org/10.1111/lnc3.12228.

[2] B. Pang, L. Lee, Opinion mining and sentiment analysis: Foundations and trends in information retrieval, Found. Trends Inf. Retr. 2 (1–2) (2008) 1–135, http://dx.doi.org/10.1561/1500000011, arXiv:1703.04009.

[3] B. Pang, L. Lee, Opinion mining and sentiment analysis, Found. Trends Inf. Retr. 1 (2) (2006) 91–231, http://dx.doi.org/10.1561/1500000001.

[4] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (1) (2012) 1–167, http://dx.doi.org/10.2200/S00416ED1V01Y201204HLT016.

[5] X. Fang, J. Zhan, Sentiment analysis using product review data, J. Big Data 2 (1) (2015) 5, http://dx.doi.org/10.1186/s40537-015-0015-2, URL: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2.

[6] T. Mikolov, G. Corrado, K. Chen, J. Dean, Efficient estimation of word representations in vector space, in: Proceedings of the International Conference on Learning Representations, ICLR-2013, 2013.

[7] T. Mikolov, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of Advances in Neural Information Processing Systems, NIPS-2013, 2013, pp. 3111–3119.

[8] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP-2014, Association for Computational Linguistics, Stroudsburg, PA, USA, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162, arXiv:1504.06654, URL: http://aclweb.org/anthology/D14-1162.

[9] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP-2014, Association for Computational Linguistics, Stroudsburg, PA, USA, 2014, pp. 1746–1751, http://dx.doi.org/10.3115/v1/D14-1181, URL: http://aclweb.org/anthology/D14-1181.

[10] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI-2015, AAAI'15, AAAI Press, 2015, pp. 2267–2273.

[11] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in: NIPS 2014 Workshop on Deep Learning, 2014, arXiv:1412.3555.

[12] M. Zulqarnain, R. Ghazali, Y.M. Mohmad Hassim, M. Rehan, Text classification based on gated recurrent unit combines with support vector machine, Int. J. Electr. Comput. Eng. (IJECE) 10 (4) (2020) 3734, http://dx.doi.org/10.11591/ijece.v10i4.pp3734-3742, URL: http://ijece.iaescore.com/index.php/IJECE/article/view/20495.

[13] J. Wang, B. Peng, X. Zhang, Using a stacked residual LSTM model for sentiment intensity prediction, Neurocomputing 322 (17) (2018) 93–101, http://dx.doi.org/10.1016/j.neucom.2018.09.049.

[14] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735, URL: https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735.

[15] Z. Lin, M. Feng, C.N. dos Santos, M. Yu, B. Xiang, B. Zhou, Y. Bengio, A structured self-attentive sentence embedding, 2017, http://dx.doi.org/10.1109/CVPR.2016.90, arXiv preprint arXiv:1703.03130.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, Nips-2017, 2017, pp. 5598–6008, http://dx.doi.org/10.1017/S0140525X16001837, arXiv:1706.03762.

[17] C. Du, H. Sun, J. Wang, Q. Qi, J. Liao, C. Wang, B. Ma, Investigating capsule network and semantic feature on hyperplanes for text classification, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 456–465, http://dx.doi.org/10.18653/v1/D19-1043, URL: https://www.aclweb.org/anthology/D19-1043.

[18] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, S. Zhang, Investigating capsule networks with dynamic routing for text classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP-2018, 2019, pp. 3110–3119, http://dx.doi.org/10.18653/v1/d18-1350, URL: arXiv:1804.00538.

[19] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT-2016, Association for Computational Linguistics, Stroudsburg, PA, USA, 2016, pp. 1480–1489, http://dx.doi.org/10.18653/v1/N16-1174, URL: http://aclweb.org/anthology/N16-1174.

[20] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, 2020, arXiv preprint arXiv:2003.08271.

[21] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the NAACL-HLT 2018, 2018, pp. 2227–2237, http://dx.doi.org/10.18653/v1/N18-1202.

[22] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, 2019, pp. 4171–4186.

[23] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, 2019, arXiv preprint arXiv:1909.11942.

[24] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, arXiv preprint arXiv:1907.11692.

[25] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2019, arXiv preprint arXiv:1911.02116.

[26] C. Sun, X. Qiu, Y. Xu, X. Huang, How to fine-tune BERT for text classification? in: China National Conference on Chinese Computational Linguistics, 2019, pp. 194–206, http://dx.doi.org/10.1007/978-3-030-32381-3_16, arXiv:1905.05583.

[27] D. Tang, B. Qin, T. Liu, Learning semantic representations of users and products for document level sentiment classification, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, ACL-2015, 2015, pp. 1014–1023, http://dx.doi.org/10.1038/srep11868, arXiv:0912.4547, URL: https://www.aclweb.org/anthology/P15-1098.

[28] M. Yang, Q. Qu, J. Zhu, Y. Shen, Z. Zhao, Cross-domain aspect/sentiment-aware abstractive review summarization, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2018, pp. 1531–1534, http://dx.doi.org/10.1145/3269206.3269273, URL: https://dl.acm.org/doi/10.1145/3269206.3269273.

[29] Y. Suhara, X. Wang, S. Angelidis, W.-C. Tan, Opiniondigest: A simple framework for opinion summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL-2020, Association for Computational Linguistics, Stroudsburg, PA, USA, 2020, pp. 5789–5798, http://dx.doi.org/10.18653/v1/2020.acl-main.513, arXiv:2005.01901, URL: https://www.aclweb.org/anthology/2020.acl-main.513.

[30] J. Ni, J. McAuley, Personalized review generation by expanding phrases and attending on aspect-aware representations, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL-2018, 2018, pp. 706–711, http://dx.doi.org/10.18653/v1/p18-2112, URL: https://www.aclweb.org/anthology/P18-2112.pdf.

[31] L. Dong, S. Huang, F. Wei, M. Lapata, M. Zhou, K. Xu, Learning to generate product reviews from attributes, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL-2017, vol. 1, 2017, pp. 623–632, http://dx.doi.org/10.18653/v1/e17-1059, URL: https://www.aclweb.org/anthology/E17-1059.pdf.

[32] Y. Koren, R. Bell, Advances in collaborative filtering, in: Recommender Systems Handbook, Springer US, Boston, MA, 2011, pp. 145–186, http://dx.doi.org/10.1007/978-0-387-85820-3_5, URL: http://link.springer.com/10.1007/978-0-387-85820-3{_}5.

[33] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2016, pp. 353–362, http://dx.doi.org/10.1145/2939672.2939673, URL: https://dl.acm.org/doi/10.1145/2939672.2939673.

[34] Y. Tay, S. Zhang, L.A. Tuan, S.C. Hui, Self-attentive neural collaborative filtering, 2018, arXiv preprint arXiv:1806.06446.

[35] H. Chen, M. Sun, C. Tu, Y. Lin, Z. Liu, Neural sentiment classification with user and product attention, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP-2016, 2016, pp. 1650–1659, URL: https://www.aclweb.org/anthology/D16-1171.pdf.

[36] Z. Wu, X.Y. Dai, C. Yin, S. Huang, J. Chen, Improving review representations with user attention and product attention for sentiment classification, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18, New Orleans, Louisiana, USA, 2018, pp. 5989–5996, arXiv:1801.07861, URL: https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16728/16166.

[37] D. Ma, S. Li, X. Zhang, H. Wang, X. Sun, Cascading multiway attention for document-level sentiment classification, in: Proceedings of the the 8th International Joint Conference on Natural Language Processing, IJCNLP-2017, 2017, pp. 634–643, URL: http://aclweb.org/anthology/I17-1064.

[38] B. Sarwar, G. Karypis, J. Konstan, J. Reidl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the Tenth International Conference on World Wide Web - WWW '01, ACM Press, New York, New York, USA, 2001, pp. 285–295, http://dx.doi.org/10.1145/371920.372071, URL: http://portal.acm.org/citation.cfm?doid=371920.372071.

[39] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2017, pp. 173–182, http://dx.doi.org/10.1145/3038912.3052569, URL: https://dl.acm.org/doi/10.1145/3038912.3052569.

[40] T. Huang, Z. Zhang, J. Zhang, Fibinet: Combining feature importance and bilinear feature interaction for click-through rate prediction, in: 13th ACM Conference on Recommender Systems, RecSys-2019, 2019, pp. 169–177, http://dx.doi.org/10.1145/3298689.3347043, arXiv:arXiv:1905.09433v1.

[41] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, URL: http://www.deeplearningbook.org.

[42] E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco, Affective computing and sentiment analysis, in: A Practical Guide To Sentiment Analysis, 2017, pp. 1–10, http://dx.doi.org/10.1007/978-3-319-55394-8_1, URL: http://link.springer.com/10.1007/978-3-319-55394-8{_}1.

[43] E. Cambria, S. Poria, A. Gelbukh, M. Thelwall, Sentiment analysis is a big suitcase, IEEE Intell. Syst. 32 (6) (2017) 74–80, http://dx.doi.org/10.1109/MIS.2017.4531228, URL: http://ieeexplore.ieee.org/document/8267597/.

[44] J. Wang, L.C. Yu, K.R. Lai, X. Zhang, Community-based weighted graph model for valence-arousal prediction of affective words, IEEE/ACM Trans. Audio Speech Lang. Process. (2016) http://dx.doi.org/10.1109/TASLP.2016.2594287.

[45] J. Wang, L.-C. Yu, K.R. Lai, X. Zhang, Investigating dynamic routing in tree-structured LSTM for sentiment analysis, in: The Conference on Empirical Methods in Natural Language Processing & International Joint Conference on Natural Language Processing, EMNLP-IJCNLP-19, 2019, pp. 3430–3435, http://dx.doi.org/10.18653/v1/d19-1343.

[46] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in: Proceedings of the 31st Conference on Neural Information Processing Systems, NIPS-2017, 2017, pp. 3859–3869, arXiv:1710.09829.

[47] A. Adhikari, A. Ram, R. Tang, J. Lin, Rethinking complex neural network architectures for document classification, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, 2019, pp. 4046–4051, http://dx.doi.org/10.18653/v1/n19-1408.

[48] M.E. Basiri, S. Nemati, M. Abdar, E. Cambria, U.R. Acharya, ABCDM: An attention-based bidirectional CNN-rnn deep model for sentiment analysis, Future Gener. Comput. Syst. 115 (2021) 279–294, http://dx.doi.org/10.1016/j.future.2020.08.005, URL: https://linkinghub.elsevier.com/retrieve/pii/S0167739X20309195.

[49] J. Wei, J. Liao, Z. Yang, S. Wang, Q. Zhao, Bilstm with multi-polarity orthogonal attention for implicit sentiment analysis, Neurocomputing 383 (2020) 165–173, http://dx.doi.org/10.1016/j.neucom.2019.11.054, URL: https://linkinghub.elsevier.com/retrieve/pii/S092523121931656X.

[50] A. Mohammadi, A. Shaverizade, Ensemble deep learning for aspect-based sentiment analysis, Int. J. Nonlinear Anal. Appl. 12 (2021) 29–38, URL: http://journals.semnan.ac.ir/article{_}4769{_}6a9264bbfcdd9d41f28a1ecfb3f8241d.pdf.

[51] M.S. Akhtar, A. Ekbal, E. Cambria, How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes], IEEE Comput. Intell. Mag. 15 (1) (2020) 64–75, http://dx.doi.org/10.1109/MCI.2019.2954667, URL: https://ieeexplore.ieee.org/document/8956109/.

[52] E. Cambria, Y. Li, F.Z. Xing, S. Poria, K. Kwok, Senticnet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, ACM, New York, NY, USA, 2020, pp. 105–114, http://dx.doi.org/10.1145/3340531.3412003, URL: https://dl.acm.org/doi/10.1145/3340531.3412003.

[53] A. Adhikari, A. Ram, R. Tang, J. Lin, DocBERT: BERT for Document Classification, 2019, arXiv preprint arXiv:1904.08398.

[54] R. Pappagari, P. Żelasko, J. Villalba, Y. Carmiel, N. Dehak, Hierarchical transformers for long document classification, 2019, arXiv preprint arXiv:1910.10781.

[55] P. Michel, G. Neubig, Extreme adaptation for personalized neural machine translation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL-2018, vol. 2, 2018, pp. 312–318, http://dx.doi.org/10.18653/v1/p18-2050, arXiv:1805.01817, URL: https://www.aclweb.org/anthology/P18-2050.pdf.

[56] R.K. Amplayo, Rethinking attribute representation and injection for sentiment classification, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 5601–5612, http://dx.doi.org/10.18653/v1/D19-1562, arXiv:1908.09590, URL: https://www.aclweb.org/anthology/D19-1562.

[57] M. Schuster, K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681, http://dx.doi.org/10.1109/78.650093, URL: http://ieeexplore.ieee.org/document/650093/.

[58] Q. Diao, M. Qiu, C.-Y. Wu, A.J. Smola, J. Jiang, C. Wang, Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS), in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2014, ACM Press, New York, New York, USA, 2014, pp. 193–202, http://dx.doi.org/10.1145/2623330.2623758, URL: http://dl.acm.org/citation.cfm?doid=2623330.2623758.

[59] L. Prechelt, Automatic early stopping using cross validation: quantifying the criteria, Neural Netw. 11 (4) (1998) 761–767, http://dx.doi.org/10.1016/S0893-6080(98)00010-0, URL: https://linkinghub.elsevier.com/retrieve/pii/S0893608098000100.

[60] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[61] S. Thrun, Multitask learning, Econ. Outlook 19 (4) (1995) 46–47, http://dx.doi.org/10.1111/j.1468-0319.1995.tb00042.x, URL: http://doi.wiley.com/10.1111/j.1468-0319.1995.tb00042.x.

You Zhang is a Ph.D. candidate in the School of Information Science and Engineering, Yunnan University, China. He received the B.S. degree in Computer Science and Engineering from Beijing Jiaotong University, China. His research interests include natural language processing, text mining, and machine learning.

Jin Wang is an associate professor in the School of Information Science and Engineering, Yunnan University, China. He holds a Ph.D. in Computer Science and Engineering from Yuan Ze University, Taoyuan, Taiwan, and another Ph.D. in Communication and Information Systems from Yunnan University, Kunming, China. His research interests include natural language processing, text mining, and machine learning.

Xuejie Zhang is a professor in the School of Information Science and Engineering, and Director of High-Performance Computing Center, Yunnan University, China. He received his Ph.D. in Computer Science and Engineering from the Chinese University of Hong Kong in 1998. His research interests include high performance computing, cloud computing, and big data analytics.