International Journal of Machine Learning and Cybernetics (2019) 10:3521–3537
https://doi.org/10.1007/s13042-019-00942-5

ORIGINAL ARTICLE

Word-character attention model for Chinese text classification

Xue Qiao[1] · Chen Peng[1] · Zhen Liu[1] · Yanfeng Hu[1]

Received: 12 September 2018 / Accepted: 18 February 2019 / Published online: 26 February 2019

Abstract
Recent progress in applying neural networks to image classification has motivated the exploration of their applications to text classification tasks. Unlike the majority of these researches devoting to English corpus, in this paper, we focus on Chinese text, which is more intricate in semantic representations. As the basic unit of Chinese words, character plays a vital role in Chinese linguistic. However, most existing Chinese text classification methods typically regard word features as the basic unit of text representation but ignore the beneficial performance of character features. Besides, existing approaches compress the entire word features into a semantic representation, without considering attention mechanism which allows for capturing salient features. To tackle these issues, we propose the *word-character attention model* (WCAM) for Chinese text classification. This WCAM approach integrates two levels of attention models: *word-level attention model* captures salient words which have closer semantic relationship to the text meaning, and *character-level attention model* selects discriminative characters of text. Both are jointly employed to learn representation of texts. Meanwhile, the word-character constraint model and character alignment are introduced in our proposed approach to ensure the highly representative of selected characters as well as enhance their discrimination. Both are jointly employed to exploit the subtle and local differences for distinguishing the text classes. Extensive experiments on two benchmark datasets demonstrate that our WCAM approach achieves comparable or even better performance than the state-of-the-art methods for Chinese text classification.

Keywords Chinese text classification · Attention mechanism · Word-character attention · Word-character constraint

1 Introduction

Previously, main approaches of text classification focus on text representations and classification methods. A number of traditional methods are applied to text classification, including Naive Bayes model [1], k-nearest neighbors algorithm [2], expectation maximization algorithm [3], support vector machine (SVM) model [4], and back-propagation neural networks [5]. However, the difficulty of feature engineering [6, 7] is regarded as a challenge in traditional text classification.

In recent years, thanks to the rapid development of deep learning methods and artificial intelligence, a lot of remarkable results have been achieved in Chinese text classification [8]. Different from traditional Chinese text classification approaches, deep learning methods are proposed to learn word embeddings [9] by deep neural network models and to perform composition over the word embeddings for Chinese text classification. For example, Zhang et al. [10] filter a vast amount of human-computer conversations to build a large-scale annotated Chinese sentiment dataset and conduct thorough experiments for sentiment classification with deep convolutional neural network (CNN) model. Zhuang et al. [11] propose a new method for Chinese text classification, which digs a basic feature of strokes to learn representation of Chinese character. Therefore, deep learning methods are proved successful in modelling sequence data like text, and thus enable to effectively improve the performance of Chinese text classification.

For some languages like Chinese, the character is the basic independent semantic unit of the word, from which the semantic meaning of the word can be inferred [12]. From the perspective of semantic analysis, words and characters have important significance for Chinese text classification. That is to say, several words or characters may determine the class of a Chinese text. For example, in news titles the words "科学家 (scientist)", "飞船 (spacecraft)", "宇宙 (universe)" occur relatively frequently in science, and the words

✉ Xue Qiao
xqiao@mail.ie.ac.cn

[1] Institute of Electronics, Chinese Academy of Sciences, Suzhou, Suzhou 215123, China

"篮球 (basketball)", "排球 (volleyball)", "体育 (sport)" are relatively unique for sports. Similarly, in user review the characters "好 (great)", "赞 (awesome)", "坏 (bad)" may show the sentiment directly. However, most existing Chinese text classification works [13, 14] typically regard word features as the basic unit of text representation but ignore the beneficial performance of character features. That is the *first limitation*.

To address the above problem, some works begin to focus on leveraging both the character-level and word-level features for achieving promising performance. Yang et al. [15] combine character embeddings, which are learned in position-based and clustered-based fashions, into sentence vectors and incorporate the compositional sentence-level representation into a neural network approach for review aspect classification. Zhang et al. [16] propose the use of character-level Convolutional Networks for text classification on Chinese news corpus. Zhou et al. [17] propose the Hybrid Attention Networks for Chinese short text classification which combines the word- and character-level selective attentions. However, when they combine the word-level and character-level features, the relationships between word and its characters as well as among these characters have not been considered, but both of them are highly helpful to find the subtle and local features. This causes the selected characters have large repetitiveness with each other which leads to redundant information. This is the *second limitation*.

To address the above two limitations, we propose the word-character attention model (WCAM) for Chinese text classification. Our WCAM approach incorporates two levels of attention models to capture salient words and discriminative characters, and further exploits the word-character constraint model and character alignment to highlight discrimination as well as eliminate redundancy of the discriminative characters. It is worthwhile to summarize the main novelties and contributions of our WCAM approach as the following aspects:

(1) *Word-character attention model* Most existing Chinese text classification works ignore the beneficial performance of character features. To address this problem, we propose the word-character attention model to combine word-level features with character-level features for improving the performance of Chinese text classification effectively. Our model integrates two levels of attention models: (1) *Word-level attention model* introduces attention mechanism in bidirectional gated recurrent unit (GRU) network [18] to capture salient words for generating text representation, which is to learn word features of text. (2) *Character-level attention model* selects the discriminative characters based on the cluster patterns of neural network, which is to learn subtle and local features of text. The word-level attention model focuses on the representative word features, and the character-level attention model focuses on the distinguishing character features among classes. Both of them are jointly applied to extract the feature of text, and enhance their mutual promotions to achieve good performance for Chinese text classification.

(2) *Word-character constraint model* Although, some works begin to focus on exploiting character features for Chinese text classification, they ignore the relationships between word and its character as well as among these characters, both of which are significate to select discriminative character. To address this problem, we propose the character selection approach driven by word-character constraint model, which highlights the saliency of characters and enhances their discrimination to ensure that the selected characters are highly representative. The word-character constraint model not only significantly promotes discriminative character selection by exploiting subtle and local distinction, but also achieves a notable improvement on Chinese text classification.

The comprehensive experimental results on two widely-used datasets demonstrate that our WCAM approach achieves comparable or even better performance than the state-of-the-art methods for Chinese text classification.

2 Related work

Most traditional methods for text classification follow the strategy of extracting basic low-level features like term frequency-inverse document frequency (TF-IDF) [19], and then generating vectors of weights for text representation [20]. However, the performance of these methods is limited by the handcrafted features. Deep learning has shown its strong power in feature learning, and achieved great progresses in text classification [21–23]. The major contributions done in the field of text classification are presented in Table 1. The comparison of the existing approaches with our WCAM approach is also given in the table. These methods can be simply divided into two groups: ensemble of networks based methods and attention based methods.

2.1 Ensemble of networks based methods

Ensemble of networks based methods are proposed to utilize multiple neural networks to learn different representations of text for better classification performance. These methods typically use distributed word representations (i.e. word embedding) as inputs to neural network models. Kalchbrenner et al. [24] propose a novel network architecture based on CNN, called dynamic convolutional neural network (DCNN), which uses dynamic k-max pooling to explicitly capture short and long-range relations without relying on a parse tree. Then Kim [25] proposes a simple CNN with two channels of word vectors which allow the

Table 1 Comparison of major works in the field of text classification

Title	Contribution	Combination of word and character features	Limitation
A convolutional neural network for modelling sentences [24]	Proposing a dynamic convolutional neural network which used dynamic k-max pooling to capture short and long-range relations	No	1. No character features 2. No attention mechanisms
Convolutional neural networks for sentence classification [25]	Proposing a simple CNN which can use both dynamic-updated and static word embeddings	No	
Recurrent convolutional neural networks for text classification [13]	Proposing a recurrent convolutional neural network for text classification which captures contextual information with the recurrent structure and constructs the representation of text using a CNN	No	
Multi-timescale long short-term memory neural network for modelling sentences and documents [23]	Introducing a multi-timescale long short-term memory neural network, a generalization of LSTMs to capture the information with different timescales	No	
Hierarchical attention networks for document classification [26]	Proposing a neural architecture, the hierarchical attention network that is designed to capture two basic insights at the word and sentence-level about document structure	No	1. No character features
Component-enhanced chinese character embeddings [27]	Developing two component-enhanced Chinese character embedding models and their bi-gram extensions for text classification on Chinese news titles	No	1. No word features 2. No attention mechanisms
Compositional recurrent neural networks for chinese short text classification [28]	Proposing a hybrid model for Chinese short text classification, which incorporates character-level into word-level features with long short-term memory networks	Yes	1. No attention mechanisms
Hybrid attention networks for Chinese short text classification [17]	Proposing the hybrid attention networks for Chinese short text classification which combines the word- and character-level selective attentions	Yes	1. No relationships between word and its characters as well as among these characters
Hierarchical convolutional attention networks for text classification [29]	Proposing a hierarchical convolutional attention networks which combines self-attention mechanisms with convolutional filters and a hierarchical structure to create a document classification model that is both highly accurate and fast to train	No	1. No character features
Character-level convolutional networks for text classification [16]	Proposing the use of character-level convolutional networks for text classification on Chinese news corpus	No	1. No word features 2. No attention mechanisms

using of dynamic-updated and static word embeddings simultaneously. However, above neural network methods are time consuming and not enough to capture the complete semantics of Chinese texts. For Chinese text classification, Lai et al. [13] apply a recurrent structure to capture contextual information when learning word representations, which may introduce considerably less noise compared to traditional window-based neural networks. Finally, a max-pooling layer is employed to capture the key components in texts for the final classification. Despite achieving promising results, these methods are still limited by ignoring the beneficial performance of character features in the Chinese text classification. Therefore, Zhang et al. [16] propose the use of character-level convolutional networks (ConvNets) for text classification on Chinese news corpus. Li et al. [27] develop two component-enhanced Chinese character embedding models and their bi-gram extensions for text classification on Chinese news titles. Zhou et al. [28] propose a hybrid model for Chinese short text classification, which incorporates character-level into word-level features with long short-term memory (LSTM) networks. However, it is difficult for these methods to find the relationships between word and its characters as well as among these characters, both of which are highly helpful for finding the subtle and local features of the Chinese text.

2.2 Attention based methods

Due to attention mechanism, we focus on the discriminative features of a text dynamically, instead of dealing with the information of entire text directly. This natural advantage makes attention mechanism widely used in many natural language processing (NLP) applications, such as machine translation [30], visual captioning [31], and question answering [32]. Inspired by recent advances in neural machine translation, Yang et al. [26] propose attention mechanism for text classification. They put forward a hierarchical structure with two levels of attention mechanisms applied at the word and sentence-level that mirrors the hierarchical structure of documents, which can attend differentially to more and less important content when constructing the document representation. Zhou et al. [33] capture the most significant semantic information in a sentence by utilizing neural attention mechanism with bidirectional LSTM networks, and propose a novel neural network called attention-based bidirectional LSTM (Att-BLSTM) for relation classification. Wang et al. [34] propose a novel convolutional neural network architecture for relation classification, which relies on two levels of attention mechanisms to better recognize patterns in complex contexts. Thus, following this elegant recipe, our WCAM approach incorporates two levels of attention models: word-level attention model focuses on the representative words, and character-level attention model focuses on the discriminative characters. Both of them are jointly employed to learn subtle and local features to enhance their mutual promotions.

3 The proposed approach

The framework of our WCAM approach is shown in Fig. 1. Firstly, the proposed approach captures salient words of text via word-level attention model to learn word features, and then it selects the discriminative characters via character-level attention model to learn the subtle and local features. *For word-level attention model*, our approach introduces attention mechanism to focus on the representative word features for better text classification performance. It utilizes the attention-based bidirectional GRU network to encode word embeddings for capturing salient word features of text, rather than only encode all the words in text that contains large semantic noise. *For character-level attention model*, our approach proposes the character selection approach driven by word-character constraint model to exploit the subtle differences among classes. Finally, we merge the prediction results of word-level attention model and character-level attention model to get the final classification result.

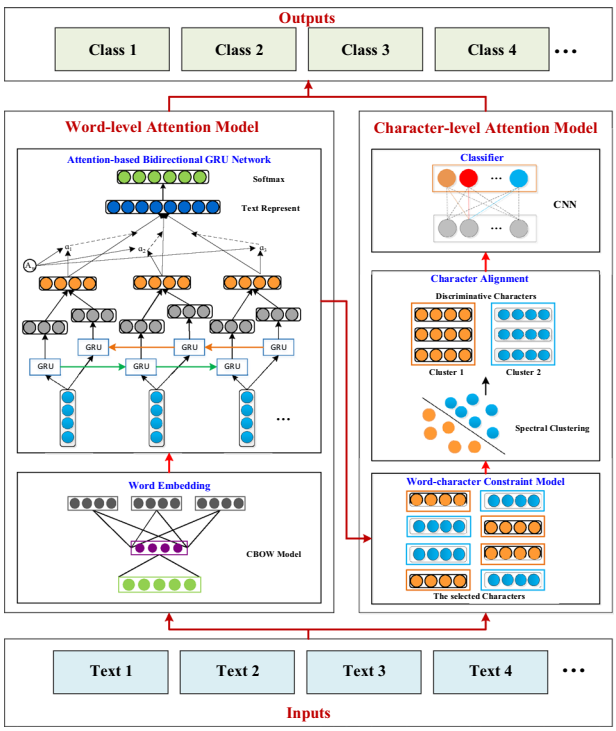

Fig. 1 Framework of the our WCAM approach

3.1 Word-level attention model

The model consists of two components: word embeddings and attention-based bidirectional GRU network. The first component is to project each word into a dense and low-dimensional vector for learning distributed representations of words, where semantically similar words are transformed into similar vector representations [35]. The second component is to capture salient words via attention-based bidirectional GRU network for generating text representation.

(1) *Word embeddings* [7]: Since pre-trained word embeddings can improve model training [36], we utilize continuous bag-of-words (CBOW) model [37] to train word embeddings with a large amount of data from Baidu Baike. The Baidu Baike is a Chinese version of Wikipedia which is closely related to hot spots and web popular.

As shown in Fig. 2, there are three-layer neural networks in the CBOW model, containing input, projection, and output layers. The CBOW model learns word embeddings by using context words to predict the center word w_d, where the context words refer to the neighboring words within a window size a near the center word in a sentence. Assume that the training dataset contains D words, and w_d with $d \in [1, D]$ represents the dth word in the training set. The CBOW model has the following objective function [7]:

$$J_{CBOW} = \max \frac{1}{D} \sum_{d=1}^{D} \sum_{-a \leqslant j \leqslant a, j \neq 0} \log p(w_d | w_{d+j}). \quad (1)$$

In the projection layer, each word w_d is embedded into an M-dimensional feature space through a word embedding matrix $W^w \in R^{D \times M}$:

$$e_d^w = W^w w_d. \quad (2)$$

Thus, the row vector of W^w in the index of w_d is the word embedding of word w_d. As a result, this component maps the input training words $\{w_1, w_2, \ldots, w_D\}$ into a series of word embeddings $\{e_1^w, e_2^w, \ldots, e_D^w\}$.

(2) *Attention-based bidirectional GRU network* [26]: In this stage, bidirectional GRU network is adopted to obtain salient words for generating text representation, as it is able to capture both past and future contextual information. Figure 3 shows the architecture of this stage. In the hidden layer, there are two GRU units that run in the opposite direction: the forward GRU unit is based on the input sequence and the backward GRU unit is based on the reverse of the input sequence. Figure 4 gives a graphical illustration of a basic GRU unit. It contains an input activation function and two types of gates (the reset gate and

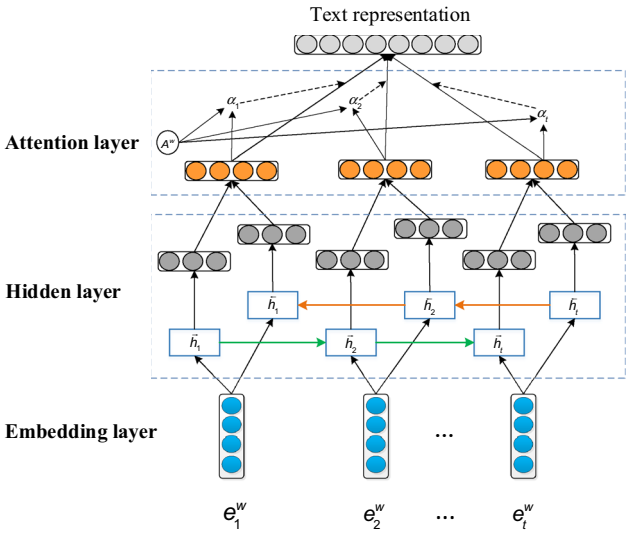

Fig. 3 Illustration of the attention-based bidirectional GRU network

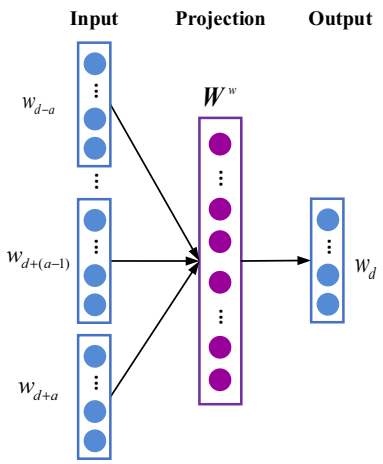

Fig. 2 Illustration of CBOW model

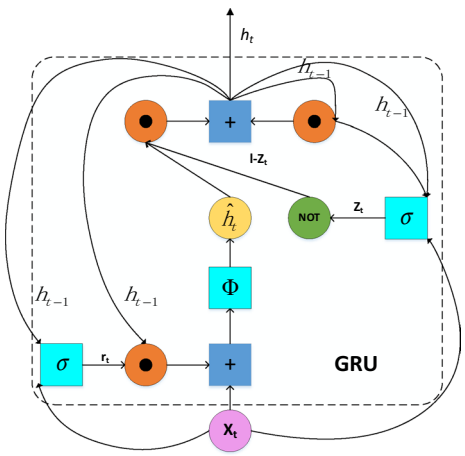

Fig. 4 Illustration of the GRU unit

the update gate), which together control how information is updated to the state. At each time step, the concatenation of the forward and backward hidden state produces the hidden state of bidirectional GRU network. Therefore, the model can capture both past and future contextual information.

In a GRU unit, the tth hidden h_t with reset gate r_t and update gate z_t is computed as:

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t, \tag{3}$$

where $\otimes$ represents the product operator. h_t is a linear interpolation between the previous state h_{t-1} and the current new state $\tilde{h}_t$ computed with new sequence information. Gate z_t decides how much past information is kept and how much new information is added. z_t is updated as:

$$z_t = \sigma\left(U^z e_t^w + W^z h_{t-1} + b_z\right) \tag{4}$$

where e_t^w is the tth word embedding, $\sigma(\cdot)$ is the logistic sigmoid function, and the variable b_z denotes the bias vector of update gate. U^z and W^z are the weight matrices of update gate. The candidate state $\tilde{h}_t$ is computed in a way similar to a traditional recurrent neural network (RNN) [38]:

$$\tilde{h}_t = \tanh(U^h e_t^w + W^h(h_{t-1} \otimes r_t) + b_h) \tag{5}$$

where $\tanh(\cdot)$ is hyperbolic tangent function and the term b_h denotes the bias vector of the state. U^h and W^h are weight matrices of the state. Here r_t is the reset gate which controls how much the past state contributes to the candidate state. If r_t is zero, then it forgets the previous state. The reset gate is updated as follows:

$$r_t = \sigma(U^r e_t^w + W^r h_{t-1} + b_r) \tag{6}$$

where the term b_r denotes the bias vector of reset gate. U^r and W^r are the weight matrices of reset gate.

Given word embeddings $\left\{e_1^w, e_2^w, \ldots, e_D^w\right\}$. The bidirectional GRU network contains the forward GRU network (i.e. $\overrightarrow{GRU}$) which reads the word embeddings from e_1^w to e_D^w and a backward GRU network (i.e. $\overleftarrow{GRU}$) which reads from e_D^w to e_1^w:

$$\overrightarrow{h}_d^w = \overrightarrow{GRU}(e_d^w),\ d \in [1, D], \tag{7}$$

$$\overleftarrow{h}_d^w = \overleftarrow{GRU}(e_d^w),\ d \in [D, 1]. \tag{8}$$

The prediction at step d is computed by concatenating the forward hidden state $\overrightarrow{h}_d^w$ and the backward hidden state $\overleftarrow{h}_d^w$, formulated as follows:

$$h_d^w = \left[\overrightarrow{h}_d^w, \overleftarrow{h}_d^w\right]. \tag{9}$$

Then, we integrate attention mechanism [39] into the outputs of hidden layer to extract salient words. Firstly, we feed h_d^w into a one-layer perceptron to obtain u_d^w as a hidden representation of h_d^w:

$$u_d^w = \tanh(W_w h_d^w + b_w), \tag{10}$$

where W_w and b_w denote the weight matrix and bias vector of this one-layer perceptron respectively. Then we measure the saliency value of word embedding e_d^w with a word attention term α_d^w, which is calculated as follows:

$$\alpha_d^w = \frac{\exp(u_d^{w^T} A^w)}{\sum_d \exp(u_d^{w^T} A^w)}, \tag{11}$$

where A^w is a randomly initialized word contextual vector [26], and it can be seen as a high-level representation over the words. After that, we compute the attentive sentence representation v^w by an attention summation of the word embeddings based on the attention term:

$$v^w = \sum_d \alpha_d^w h_d^w. \tag{12}$$

According to the word-level attention model, we can capture salient words in text to finally obtain the attentive text representation.

3.2 Character-level attention model

In Chinese linguistic, the discriminative characters are crucial for text classification, especially for distinguishing the classes with slight semantic differences. Although some works devote to exploit character features to improve Chinese text classification, they ignore the relationships between the word and its characters as well as among these characters, leading to the problem that the selected characters may have large semantic repetitiveness with each other, and some discriminative characters are ignored. Therefore, a novel character selection approach driven by character-level attention model is proposed to exploit the subtle discrimination which is used to assist the word features for better classification performance. The character-level attention model consists of two components: word-character constraint model and character alignment. The first component aims to select the discriminative characters, and the second component aims to align the selected characters into clusters by the semantic meaning.

(1) Word-character constraint model: Given a Chinese text, its salient words and the saliency value of each salient word are obtained via word-level attention model. Then, discriminate characters selection is driven by word-character constraint model as follows:

Let $\mathbb{C}$ denotes the candidate characters that segmented from salient words. That is to say, we obtain the candidate characters $\mathbb{C}$ by segmenting the salient words into characters. Let $C = \left\{c_1, c_2, ..., c_N\right\}$ denotes N characters selected from

$\mathbb{C}$ as the discriminative characters for each given text. The word-character constraint model aims to solve the following optimization problem:

$$C^* = \arg\max_{\mathbb{C}} \Delta(C) \tag{13}$$

where $\Delta(C)$ is defined as a function that jointly models saliency and the relationships among characters as follows:

$$\Delta(C) = \log\left(\left|C - C_R\right|\right) + \log\left(Mean\left(\alpha_C\right)\right) \tag{14}$$

where C_R denotes the set of repeated characters in C, and $Mean\left(\alpha_C\right)$ denotes the average saliency value of all salient words that contain the N characters, which is defined as follows:

$$Mean\left(\alpha_C\right) = \frac{1}{\left|W_C\right|}\sum_i \alpha_{w'_i}, \tag{15}$$

where W_C refers to the set of salient words that contain all the N characters, w'_i refers to the ith salient word in W_C, and $\alpha_{w'_i}$ refers to the saliency value of salient word w'_i. Word-character constraint model aims to select the most discriminative characters, which consists of two items: The first item realized by $\log\left(\left|C - C_R\right|\right)$ is to reduce the repetitiveness among selected characters, and the second item realized by $\log\left(Mean\left(\alpha_C\right)\right)$ is to maximize the saliency of selected characters. To maximize the values of both two items, we adopt the sum operation in Eq. 14.

(2) Characters alignment: The characters selected by the word-character constraint model are in disorder and not aligned by their semantic meaning. Since characters with different semantic meanings contribute differently to the final prediction, an intuitive idea is to align the characters with the same semantic meaning together.

Inspired by the fact that hidden layers of the CNN show clustering patterns, we perform character clustering on the neurons of hidden layers in the CNN to align the selected characters. Figure 5 conceptually shows what this step performs. Firstly, we compute the cosine similarity of weights between two mid-layer neurons u_i and u_j, which is denoted as the similarity matrix $S(i,j)$, and then spectral clustering [39] is performed on the similarity matrix S to partition the mid-layer neurons into m groups.

Then, we use the character clusters to align the selected characters as follows: (1) Project each selected character into a real-value vector to obtain character embeddings. (2) Feed forward the character embeddings to CNN layers to produce an activation score for each neuron. (3) Sum up the scores of neurons in one cluster to get cluster score. (4) Align the selected characters to the cluster with highest cluster score, which is formulated as follows: Given a text, the selected characters are obtained through word-character constraint model, and then character alignment is performed on these characters with m character clusters $L = \left\{l_1, l_2, ..., l_m\right\}$.

Through character-level attention model, the discriminative characters in texts are selected to train a CNN called *CharacterCNN* for obtaining the prediction of character-level attention. The structure of *CharacterCNN* is a slight variant of the CNN structure of Kim et al. [25], as shown in Fig. 6. Different from the standard CNN with several convolutional and pooling layers, our *CharacterCNN* is designed with a single convolutional layer, followed by a global average pooling layer and a fully connected layer. The number of neurons in output layer is set as the number of classes.

3.3 Text classification

For better classification performance, we train the attention-based bidirectional GRU network and CNN simultaneously to get two classifiers, called *WordGRU* and *CharacterCNN* respectively. The two classifiers are all text classifiers: *WordGRU* for salient words and *CharacterCNN* for selected discriminative characters. However, since they focus on the different levels of text, their impacts and strengths are different. The different level focuses (i.e. words of original text and characters of original text) are complementary to improve the Chinese text classification performance. Finally, the prediction results of the two different levels are merged as follows:

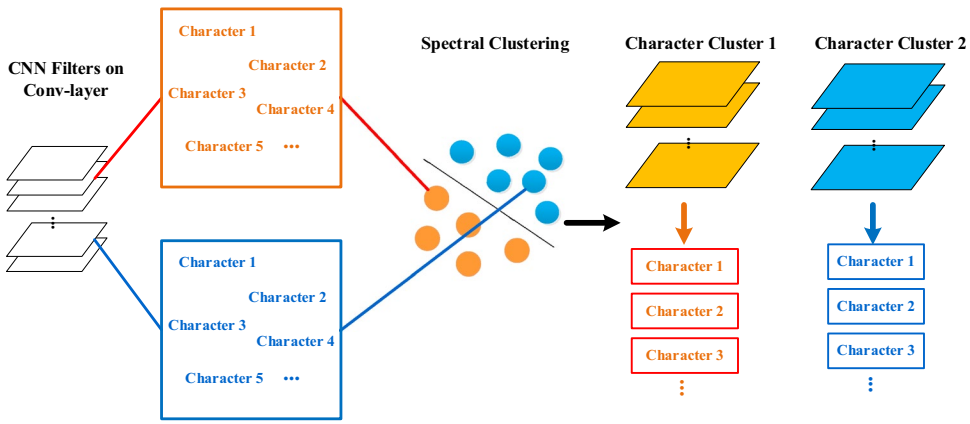

Fig. 5 Illustration of the character alignment

$$Score = \beta * word\_score + \gamma * character\_score \quad (16)$$

where *word_score* and *character_score* are the softmax values of *WordGRU* and *CharacterCNN* respectively. β and γ are selected through the k-fold cross-validation method. Finally, the class label with the highest *Score* is chosen as the classification result.

In summary, the detailed algorithm of our proposed WCAM is as follows:

Algorithm 1 The algorithm of WCAM

Input: $T = T^l \cup T^u$, where T^l is the labeled Chinese text set (i.e. training dataset) and T^u is the unlabeled Chinese text set (i.e. testing dataset).

Output: The class labels of testing dataset.

1: Segment training dataset T^l into words to obtain the input training words $\{w_1, w_2, \ldots, w_D\}$.

2: Utilize the CBOW model to pre-train word embeddings with a large amount of data from Baidu Baike. Thus, the word embeddings of training dataset (i.e. $\{e_1^w, e_2^w, \ldots, e_D^w\}$) are obtained.

3: Train the attention-based bidirectional GRU network with $\{e_1^w, e_2^w, \ldots, e_D^w\}$ to capture the salient words of training dataset as well as get the *WordGRU* classifier.

4: Segment the salient words into characters to obtain the candidate characters $\mathbb{C}$.

5: Solve Eq. 13 to obtain the discriminative characters $C = \{c_1, c_2, ..., c_N\}$.

6: Compute the cosine similarity of weights between two mid-layer neurons to obtain the similarity matrix S.

7: Perform spectral clustering on the similarity matrix S to partition the mid-layer neurons into m groups. Thus, the character clusters $L = \{l_1, l_2, ..., l_m\}$ are obtained.

8: Utilize the character clusters $L = \{l_1, l_2, ..., l_m\}$ to align the discriminative characters, and the i-th discriminative character c_i is aligned into l_c as follows:

8.1: Set $score_k = 0; k = 1, 2, ..., m.$

8.2: Project each selected character into a real-value vector to obtain character embeddings.

8.3: Perform a feed-forward pass to compute c_i's activations $F_i = \{f_{i1}, f_{i2}, ..., f_{id}\}$.

8.4: for $k = 1, 2, ..., m; j = 1, 2, ..., d$ **do**

if j-th neuron belongs to cluster l_k **then**

$score_k = score_k + f_{ij}$.

end if

end for

8.5: $c = \arg\max_k score_k$.

8.6: return l_c.

9: Train a CNN with the aligned discriminative characters to get *CharacterCNN* classifier.

10: As to testing dataset T^u, the class labels are predicted by solving Eq. 16.

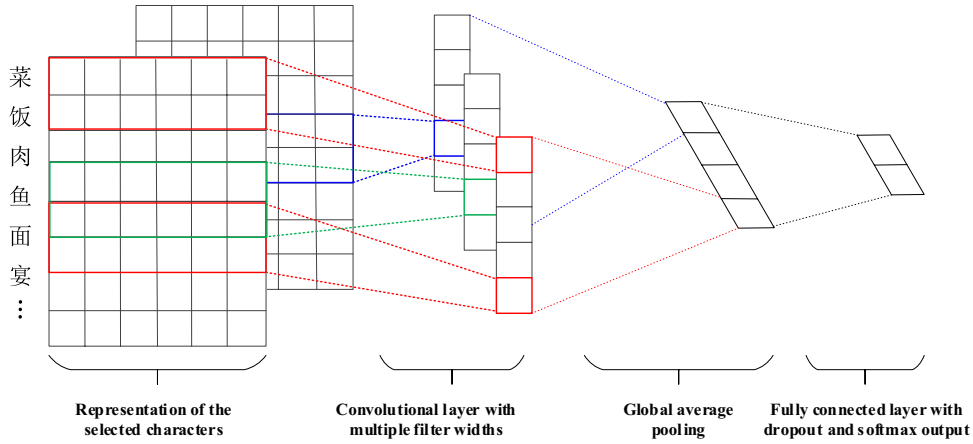

Fig. 6 The structure of *CharacterCNN*

4 Experiments

In this section, we evaluate the performances of our WCAM approach by comparing with the state-of-the-art methods. Firstly, we describe datasets and experimental settings. Then, we conduct several experiments on the task of Chinese text classification and report empirical results.

4.1 Datasets

We conduct experiments separately on two large-scale text classification datasets, THU Chinese News (THUCNews) and Chinese News Titles (CNT) [28]. The THUCNews dataset is obtained from the historical data of Sina News subscription channels from 2005 to 2011, containing 91,000 news with 14 classes (finance, lottery, estate, stock, furniture, education, science, society, fashion, politics, constellation, sport, game, entertainment). The dataset of CNT contains 59,938 titles with 30 classes. Specially, the two datasets of different lengths are selected to prove the superiority of our model in terms of text length. We split each dataset into a training set, a validation set and a testing set according to 8:1:1, respectively. The detailed information of the two datasets is summarized in Table 2.

4.2 Experimental setting

In the experiments, we use natural language processing information retrieval (NLPIR) [40], a tool for Chinese word segmentation, to segment experimental texts into words.

There are several hyper-parameters in our experiments, including the maximum length of input text l_m, the dimension of word embedding M_w, the hidden layer dimension of GRU network (i.e. M_{GRU}), the dropout probability ρ, the learning rate of stochastic gradient descent (i.e. α), the mini-batch size $B_{\min}$ and the epoch size s_{ep}.

We pick optimal hyper-parameters using the grid-search method on the validation set, of which the range of these hyper-parameters is adopted by experience. Following [41], we tune the hyper-parameters via fivefold cross-validation, and the hyper-parameter settings are reported in Table 3.

4.3 The effect of components in our WCAM approach

This sub-experiment aims to study the effect of components in the proposed approach, and the detailed experiments are performed from the following three aspects:

(1) The effect of word-level attention and character-level attention models: In our WCAM approach, the final prediction score is merged by the prediction scores of word-level

Table 2 Description of datasets

Dataset	Texts	Length (avg.)	Training	Validation	Testing
THUCNews	91,000	2339	72,800	9100	9100
CNT	59,938	18	47,950	5994	5994

Table 3 Hyper-parameter Settings

Hyper-parameters	Range	Choice	
		THUCNews	CNT
l_m	50–1000	600	70
M_w	50, 100, 200, 300	100	100
M_c	50, 100, 200, 300	100	100
M_{GRU}	100–200	128	128
s_{ep}	10	10	10
B_{min}	128	128	128
ρ	0.5, 0.6, 0.7, 0.8, 0.9	0.8	0.8
α	1, 0.1, 0.01, 0.001	0.001	0.01

Table 4 Comparisons of classification results on THUCNews and CNT datasets

Method	THUCNews			CNT		
	Precision (%)	Recall (%)	F_1 (%)	Precision (%)	Recall (%)	F_1 (%)
Original word	93.08	91.69	92.38	89.49	88.59	89.04
Original character	91.37	90.16	90.76	87.02	86.37	86.69
Original word + original character	94.42	93.19	93.8	90.39	89.62	90
Word-level	93.85	92.66	93.25	89.52	88.84	89.18
Character-level	91.84	90.7	91.27	88.11	87.18	87.64
Our WCAM approach (word-level + character-level)	95.64	94.31	94.97	91.52	90.71	91.11

attention model and character-level attention model, which are denoted as "Word-level" and "Character-level". We investigate the following methods for comparison and verify the effectiveness of word-level and character-level models.

- "Original Word": this method utilizes only the word embedding of each word as features to train bidirectional GRU network.
- "Original Character": it is same as Original Word except that word embedding is replaced by the character embedding.
- "Original Word + Original Character": this method combines "Original Word" and "Original Character". The final prediction result is generated by merging the prediction results of two different methods, i.e. "Original Word" and "Original Character".

This sub-experiment is conducted on both datasets, and precision, recall, and F1-score are adopted to evaluate the performance. The results are demonstrated in Table 4, from which we can draw the following observations:

1. For methods of "Original Word", "Original Character", "Word-level", and "Character-level", it shows that comparing with the results of "Original Word", "Word-level" improves F1-score by 0.87% and 0.14% on two datasets respectively, owing to capturing the salient words for learning representation of text. The classification results of "Character-level" are not better than "Original Word", i.e. 91.27% vs. 92.38% F1-score on THUCNews dataset and 87.64% vs. 89.04% F1-score on CNT dataset. It is probably due to that the character-level attention model focuses on subtle and local features of word, containing less information than original text. However, "Character-level" still achieves considerable classification results, which is better than "Original Character" method (i.e. 91.27% vs. 90.76% F1-score on THUCNews dataset, and 87.64% vs. 86.69% F1-score on CNT dataset).
2. For methods of "Original Word + Original Character", "Word-level", and "Character-level", it can be seen that the F1-scores of "Word-level" and "Character-level" are slightly lower than that of "Original Word + Original Character" (i.e. 93.25% and 91.27% vs. 93.80% on THUCNews dataset, and 89.18% and 87.64% vs. 90.00% on CNT dataset). This is because the "Original Word + Original Character" method includes the information not only from words but also from characters, it can provide more supplementary information than "Word-level" and "Character-level" to achieves better performance. However, combining "Word-level" and "Character-level" improves F1-score a lot than "Original Word + Original Character", i.e. by 1.17% and 1.11% on the two datasets respectively.
3. For methods of "Word-level", "Character-level", and "Word-level + Character-level", it indicates that our WCAM approach, the combination of word-level and character-level attention models, achieves better classification results than only one level attention model (i.e. 94.97% vs. 93.25% and 91.27% F1-score on THUNews dataset, and 91.11% vs. 89.18% and 87.64% F1-score on CNT dataset). It owes the different but complementary focuses of word-level and character-level attention mod-

Table 5 Comparisons of classification results on THUCNews and CNT datasets

Method	THUCNews				CNT			
	Precision (%)	Recall (%)	F_1 (%)	AUC	Precision (%)	Recall (%)	F_1 (%)	AUC
CA	87.33	84.88	86.25	0.868	77.64	80.79	79.18	0.843
WCC	93.36	89.67	92.13	0.927	82.4	86.74	84.51	0.89
WCC + CA (our WCAM approach)	95.64	94.31	94.97	0.95	91.92	90.63	91.27	0.913

els: the word-level attention model focuses on capturing representative features, while the character-level attention model focuses on mining subtle and local features among classes.
4. It is obvious that our WCAM approach is superior to all the comparative methods on the two datasets due to the complementary information between word and character.

These observations illustrate that our proposed word-level and character-level attention methods are complementary to each other, so their combination can further boost the performance of Chinese text classification.

(2) The effect of word-character constraint model and character alignment: In this study, our task is to evaluate the effect of word-character constraint as well as character alignment on our WCAM approach. The results of this sub-experiment are shown in Table 5, where "WCC" refers to the word-character constraint model, "CA" refers to character alignment, and "WCC + CA" refers to the combination of the above two methods, which is adopted by our WCAM approach. From Table 5, we can observe that:

1. For both datasets, the results of characters selected by word-character constraint model ("WCC") are better than the results of characters selected with character alignment ("CA"), e.g. 92.13% vs. 86.25% F1-score on THUCNews dataset, and 84.51% vs. 79.18% F1-score on CNT dataset.
2. In addition, applying character alignment on the basis of word-character constraint model further improves the classification performance, as indicated from the results that "WCC + CA" outperforms the other two methods.

Furthermore, the area under curve (AUC) [42] is introduced to evaluate the performance in more detail, since it can reflect the relationship between the specificity (false negative rate) and the sensitivity (true positive rate) in classification. Figure 7 summarizes the quality of the individual class predictors in terms of AUC values on the two datasets respectively. The results show that: For "WCC + CA" on both datasets, each class can be predicted better than random (the AUC of random prediction is 0.5) on average. The performance of our WCAM approach is appreciative for most classes. But in some classes, the AUC values of our WCAM approach do not provide a more accurate prediction than other methods (e.g. classes "estate", "furniture" and "education" in THUCNews dataset, and "baby", "comic", "digi", "food", "game" and "house" in CNT dataset). This effect can be explained as follows: On the one hand, the lengths of classes such as "estate", "furniture" and "education" in THUCNews dataset are too long that a large number of ambiguous characters are generated to reduce the prediction quality. On the other hand, as the number of classes in each dataset increases, it is more difficult to achieve perfect prediction for all classes. However, for our WCAM approach, average higher accuracies are also achieved on both datasets than other methods in comparison, as indicated by the mean AUC in Table 5.

Therefore, the results verify that aligning discriminative characters with the same semantic meaning together can assist word-character constraint model to further improve the results of word-character attention model.

4.4 Comparison to the state-of-the-art methods

In this sub-experiment, we present further insight into evaluating the performance of our proposed WCAM approach, through comparing it with the following state-of-the-art methods.

- SVM + TF-IDF: This approach trains SVM classifier with LIBLINEAR by utilizing the counts of TF-IDF for each word as features.
- SVM + Average E_w: It is same as SVM + TF-IDF except that TF-IDF is replaced by the average word embeddings (Average E_w). In detail, we learn 100-dimensional word

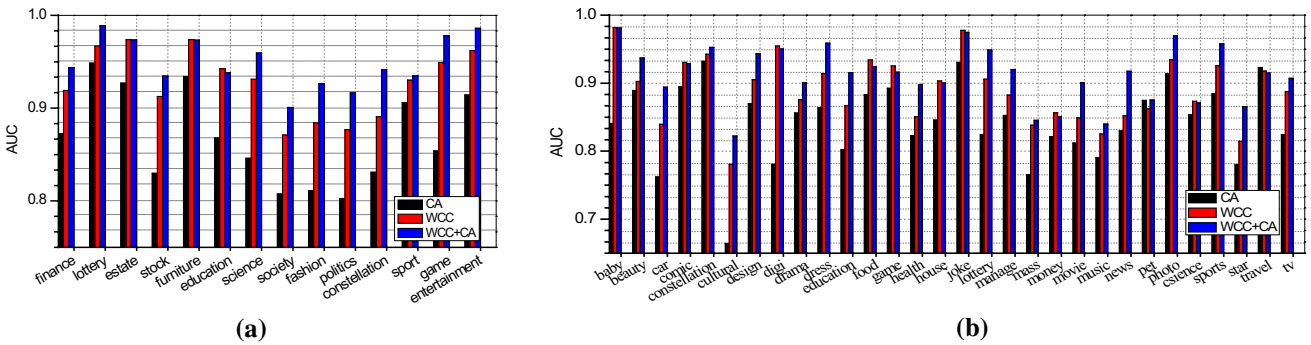

Fig. 7 Comparison of individual class predicting quality. **a** THUCNews dataset. **b** CNT dataset

embeddings with word2vec [43], which are averaged to get text representation, and train the SVM classifier.

- In CNN-based methods (CNN + E_w, CNN + E_w + E_c and Att-CNN), we employ the CNN model reported in [25] and conduct experiments with three inputs respectively. In detail, CNN + E_w and CNN + E_w + E_c use word embeddings (+ E_w) and the combination of word embeddings and character embeddings (+ E_w + E_c) as inputs respectively. For Att-CNN, a weighted attention mechanism is introduced to learn weights of specific information to enhance classification.
- Same as CNN-based methods above, other similar methods are proposed, i.e. RNN-based methods (RNN + E_w, RNN + E_w + E_c and Att-RNN), LSTM-based methods (LSTM + E_c, LSTM + E_w + E_c and Att-LSTM), BLSTM-based methods (BLSTM + E_w, BLSTM + E_w + E_c and Att-BLSTM) and GRU-based methods (GRU + E_w, GRU + E_w + E_c and Att-GRU). In RNN-based methods, we employ the RNN model reported in [44] and experiment with three inputs respectively, which are the same as CNN. In LSTM-based methods, we implement a single layer recurrent neural network with the common "vanilla" LSTM [45], of which the output dimension is 512. In BLSTM-based methods, we implement a recurrent neural network with bidirectional LSTM and conduct experiments with three inputs respectively, which are the same as RNN and LSTM.
- C-LSTMs [28]: Based on LSTM, this approach proposes an effective way to compose character features with word features for Chinese short text representation.
- HN-ATT [26]: This approach proposes a hierarchical attention network by introducing two levels of attention mechanisms applied at the word and sentence-level for text classification.
- HCANs [29]: This approach introduces a new self-attention-based text classification architecture that can capture linguistic relationships over long sequences.
- CA-GRUs [46]: This approach develops a new architecture termed Comprehensive Attention GRU network, which aims to store comprehensive information and local contexts in a sequence for text classification.
- HANs-BLSTM + CNN [17]: This approach proposes a novel attention network for Chinese short text classification, which aggregates important characters and words into sentence vectors respectively and merges them into one representative vector.

The full results are summarized in Table 6, from which we make the following observations:

Table 6 Comparisons with state-of-the-art methods on THUCNews and CNT datasets

Methods	THUCNews			CNT		
	Precision (%)	Recall (%)	F_1 (%)	Precision (%)	Recall (%)	F_1 (%)
SVM + average E_w	82.56	84.21	83.38	73.42	72.32	72.87
SVM + TF-IDF	83.85	82.41	83.12	74.34	72.89	73.61
CNN + E_w	85.91	84.83	85.37	76.02	75.57	75.79
CNN + E_w + E_c	86.26	85.61	85.93	77.55	76.46	77
Att-CNN	86.52	85.87	86.19	78.13	76.92	77.52
RNN + E_w	86.15	85.06	85.6	76.66	75.54	76.1
RNN + E_w + E_c	87.36	86.25	86.8	77.82	76.61	77.21
Att-RNN	88.06	86.75	87.4	78.83	77.62	78.22
LSTM + E_w	89.85	88.3	89.07	80.18	78.76	79.46
LSTM + E_w + E_c	90.74	88.96	89.84	80.85	79.12	79.98
Att-LSTM	91.03	89.05	90.03	81.25	79.28	80.25
BLSTM + E_w	90.11	88.88	89.49	80.36	79.09	79.72
BLSTM + E_w + E_c	91.25	90.12	90.68	81.52	80.36	80.94
Att-BLSTM	91.87	90.58	91.22	82.23	80.89	81.55
GRU + E_w	91.24	89.94	90.59	81.51	80.21	80.85
GRU + E_w + E_c	91.57	90.53	91.05	82.03	80.77	81.4
Att-GRU	91.86	90.44	91.14	82.21	80.85	81.52
C-LSTMs	92.55	91.28	91.91	82.44	81.12	81.77
HN-ATT	93.11	91.77	92.44	85.92	84.63	85.27
HCANs	93.35	92.09	92.72	86.15	84.98	85.56
CA-GRUs	94.37	92.79	93.57	90.56	89.86	90.21
HANs-BLSTM + CNN	95.08	93.61	94.34	90.85	90.13	90.49
WCAM approach	95.64	94.31	94.97	91.92	90.63	91.27

1. In comparison to CNN + E_w, RNN + E_w, LSTM + E_w, BLSTM + E_w and GRU + E_w, these results indicate that methods with the combination inputs (i.e. CNN + E_w + E_c, RNN + E_w + E_c, LSTM + E_w + E_c, BLSTM + E_w + E_c and GRU + E_w + E_c) achieve better performance on both datasets. In detail, compared with CNN + E_w, the performance of CNN + E_w + E_c is improved by 0.56% and 1.21% in terms of F1-score on two datasets respectively. For RNN + E_w + E_c, it gets the improvement from RNN + E_w by 1.2% and 1.11% in terms of F1-score on two datasets respectively. Compared with LSTM + E_w, LSTM + E_w + E_c achieves 0.77% (THUCNEWs) and 0.52% (CNT) higher F1-score respectively. For BLSTM + E_w + E_c, it improves 1.19% (THUCNEWs) and 1.22% (CNT) F1-score than BLSTM + E_w respectively. Also, for GRU + E_w + E_c, it obtains 0.46% and 0.82% higher F1-score than GRU + E_w on two datasets respectively. This indicates the effectiveness to enhance word features with character features.
2. On both datasets, the performances of methods with attention mechanism (i.e. Att-CNN, Att-RNN, Att-LSTM, Att-BLSTM and Att-GRU) are slightly better than the corresponding methods without it, which is due to the effectiveness of attention mechanism to further capture useful semantic meaning for Chinese text classification.
3. Compared with C-LSTMs, HN-ATT, HCANs, CA-GRUs and HANs-BLSTM + CNN, our WCAM approach yields an even more brilliant performance on both datasets. On THUCNews dataset, our approach performs better with 3.06%, 2.53%, 2.25%, 1.40% and 0.63% F1-score increases than C-LSTMs, HN-ATT, HCANs, CA-GRUs and HANs-BLSTM + CNN respectively. In addition, on CNT, our approach also achieves better F1-score than C-LSTMs (91.27% vs. 81.77%), HN-ATT (91.27% vs. 85.27%), HCANs (91.27% vs. 85.56%), CA-GRUs (91.27% vs. 90.21%) and HANs-BLSTM + CNN (91.27% vs. 90.49%) respectively.
4. Obviously, our WCAM approach outperforms all of the other state-of-the-art methods for Chinese text classification task on both datasets, and it verifies the superiority of our proposed WCAM approach, which jointly integrates word-level and character-level attention models to boost feature learning and enhance their complementarity. Furthermore, our WCAM approach employs the word-character constraint model to exploit the subtle and local features for distinguishing similar classes.

Another measure of classification performance besides precision, recall and F1-score is the receiver operator characteristic (ROC) curve [42]. The closer to left upper corner of coordinator the ROC curve is, the better classification performance the model achieves. We plot the corresponding ROC curve of each method on two datasets (see Fig. 8), and there are two main observations: On one hand, for all methods, the performances of ROC curves are apparently better than random on both datasets. On the other hand, the ROC curves of our WCAM approach on both datasets outperform other comparative methods, indicating the effectiveness of our WCAM approach on Chinese text classification.

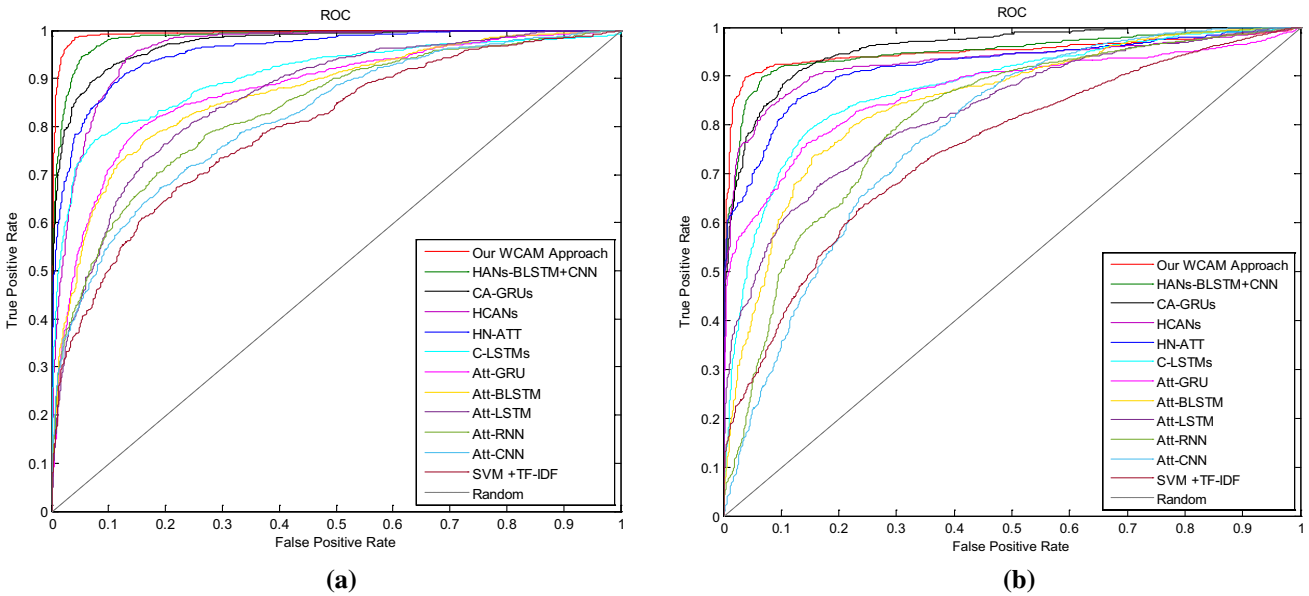

Fig. 8 ROC curves of each method. **a** THUCNews dataset. **b** CNT dataset

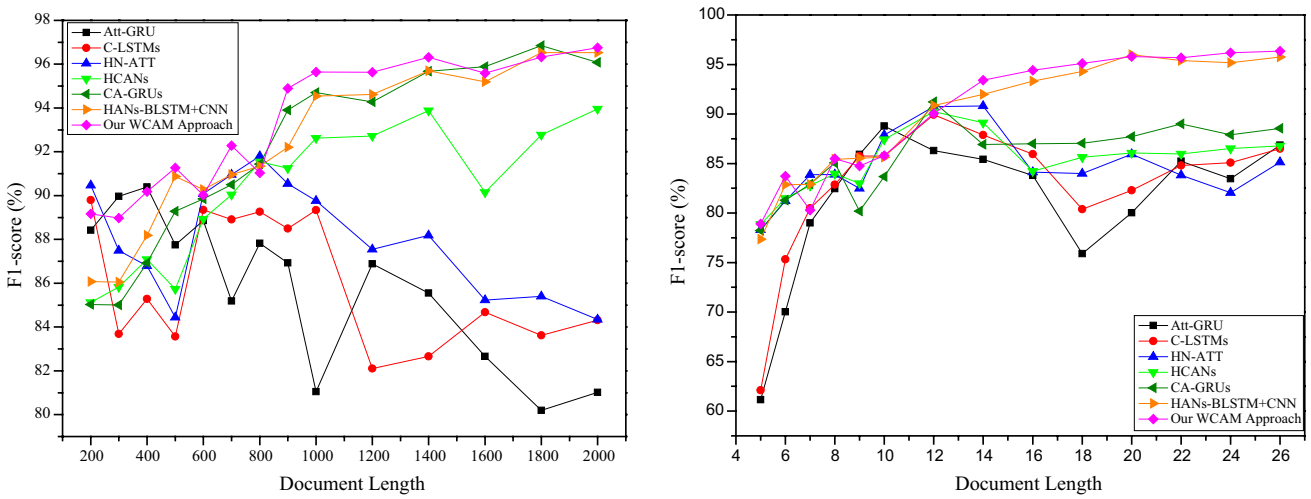

Fig. 9 The F1-score of classification with respect to the lengths of texts. **a** THUCNews dataset. **b** CNT dataset

Besides, the results of Fig. 8 coincide with the observations from Table 6.

In addition, to study the classification performance variance with different length of texts, we group texts of similar lengths and introduce F1-score to evaluate the performance of per group (see in Fig. 8), following [47]. There are a few observations from Fig. 9:

1. On both datasets, the average F1-scores of our WCAM approach (93.15% on THUCNews and 89.71% on CNT) are clearly higher than that of the comparative methods. In detail, for Att-GRU, C-LSTMs, HN-ATT, HCANs, CA-GRUs and HANs-BLSTM + CNN the average F1-score are 85.91%, 86.08%, 88.08%, 90.12%, 91.81% and 92.08% respectively on THUCNews, and 81.02%, 82.51%, 84.60%, 85.15%, 85.48% and 89.46% respectively on CNT.
2. Our WCAM approach is more robust to the length of the texts, of which the performance rises as the length of texts increases and show no deterioration even with texts of length 26 (CNT) and 2000 (THUCNews). On the contrary, the performances of Att-GRU, C-LSTMs, HN-ATT, and CA-GRUs slightly degrade as the length of texts increases on CNT sets, and for each method, the performance even dramatically drops when the length of texts is higher than a definite value on THUCNews dataset.

The observations above illustrate that our WCAM approach is more effective than other methods in handling any length of texts, which further confirms the superiority of our WCAM approach over other state-of-the-art methods for Chinese text classification.

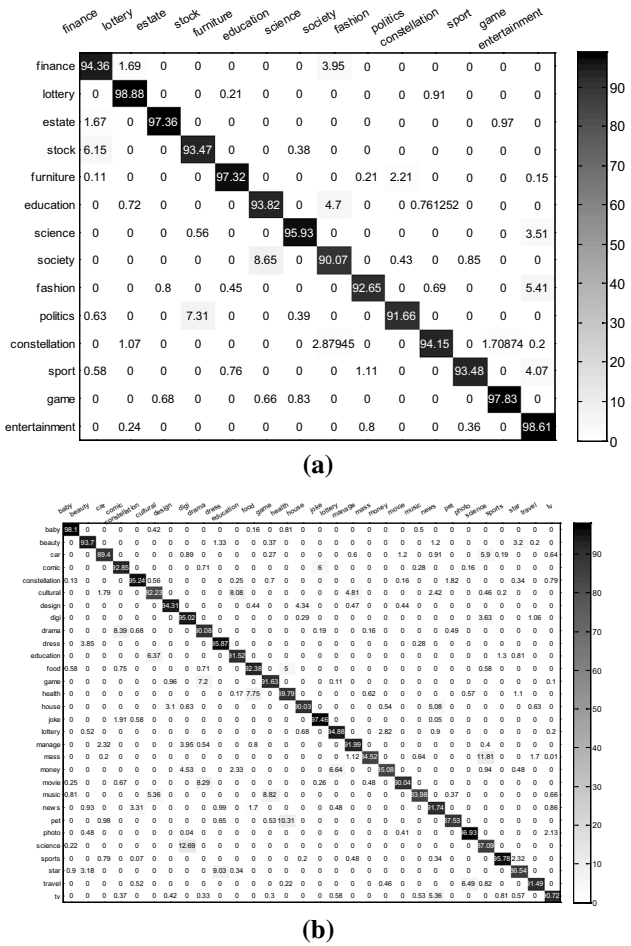

Fig. 10 Confusion matrices of classification results for WCAM approach (normalized with percentage values). **a** THUCNews dataset. **b** CNT dataset

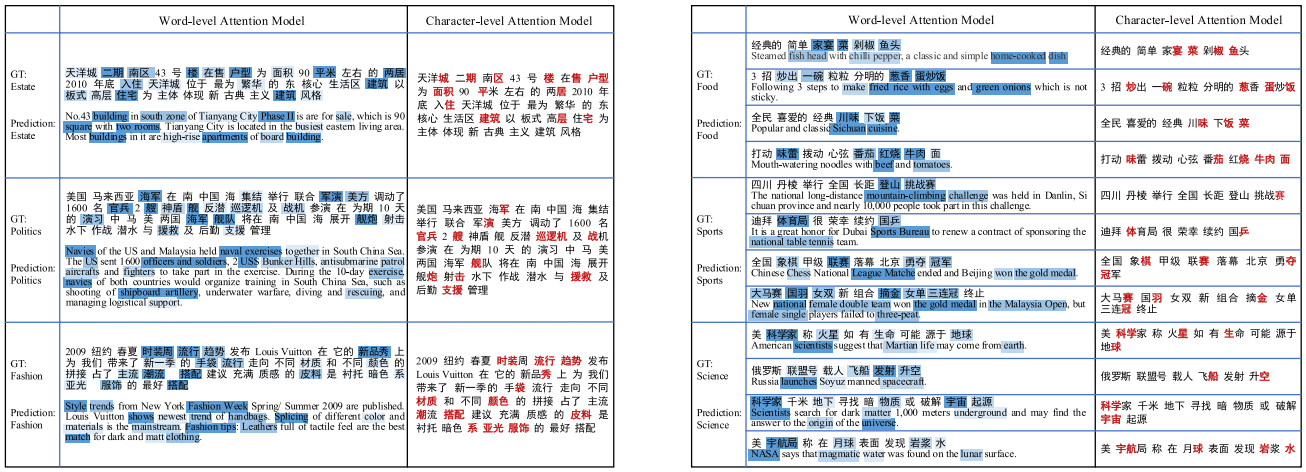

Fig. 11 Visualizations from the word-level and character-level attention model. **a** THUCNews dataset. **b** CNT dataset

Figure 10 shows the confusion matrix of the classification accuracy for our WCAM approach on both datasets. In the confusion matrix, the quality of classifier can be measured in terms of normalized multi-class accuracy on the main diagonal of the confusion matrix. From Fig. 10 the same conclusion can be drawn that our WCAM approach achieves excellent performance on Chinese text classification.

4.5 Case study

To validate that our WCAM approach is able to capture informative words and characters in a text, we provide some typical instances in both datasets for example. Specifically, we visualize the word-level attention model and character-level attention model for several texts from the THUCNews and CNT datasets, as shown in Fig. 11. The highlighted words and characters show the visualization results of word localization and character selection respectively. Blue denotes salient words of original texts via word-level attention model, and red denotes selected discriminative characters via character-level attention model. The color depth indicates importance degree of the salient words, the darker the more important. From Fig. 11, we have the following observations:

1. Word-level attention model can capture words that are closely related to text topic and contain most critical information. For example, in the first text about estate in Fig. 11a, "Phase II", "building", "square", "two rooms", "high-rise apartment", etc. are highlighted.
2. Character-level attention model can select discriminative character features and then obtain subtle and local differences for distinguishing the text classes. For example, in the first text about estate in Fig. 11a, "Phase", "rooms", "square", "apartments", etc. are highlighted. Besides, it is complementary with salient words, so their combination further boosts the classification accuracy.
3. The two different level focuses (i.e. words of original text, and characters of original text) have different representations and are complementary to improve the prediction.

5 Conclusion

In this paper, we propose the WCAM approach, which improves the performance of Chinese text classification by jointly integrating two levels of attention models: word-level attention model captures salient words in text, and character-level attention model selects discriminative characters of words. The two levels of attention models jointly improve the feature learning and enhance their mutual promotions. As a second contribution, we propose word-character constraint model and character alignment to ensure the high representativeness as well as discrimination of the selected characters. The results of extensive experiments on THUCNews dataset and CNT dataset demonstrate that our WCAM approach achieves comparable or even better performance than the state-of-the-art methods for Chinese text classification.

Acknowledgements This work was supported by Gusu Innovation Talent Foundation of Suzhou under Grant ZXT2017002 and National Key R&D Program of China under Grant 2017YFC08219.

References

1. Pratama BY, Sarno R (2016) Personality classification based on Twitter text using Naive Bayes, KNN and SVM. In: IEEE international conference on data and software engineering, pp 170–174
2. Wandabwa H, Zhang D, Sammy K (2017) Text categorization via attribute distance weighted k-nearest neighbor classification. In: IEEE international conference on information technology, pp 225–228
3. Steyn C, Waal AD (2017) Semi-supervised machine learning for textual anomaly detection. In: IEEE pattern recognition association of South Africa and robotics and mechatronics international conference, pp 1–5
4. Haddoud M, Mokhtari A, Lecroq T et al (2016) Combining supervised term-weighting metrics for svm text classification with extended term representation. Knowl Inf Syst 49(3):1–23
5. Tuteja SK, Bogiri N (2017) Email spam filtering using BPNN classification algorithm. In: IEEE international conference on automatic control and dynamic optimization techniques, pp 915–919
6. Sun RH, Hao J (2017) Comparisons of word representations for convolutional neural network: an exploratory study on tourism Weibo classification. In: IEEE international conference on service systems and service management, pp 1–5
7. Li J, Li J, Fu X et al (2016) Learning distributed word representation with multi-contextual mixed embedding. Knowl Based Syst 106(C):220–230
8. Cheng J, Li P, Ding Z et al (2017) Sentiment classification of chinese microblogging texts with global RNN. In: IEEE international conference on data science in cyberspace, pp 653–657
9. Liu S, Bremer PT, Thiagarajan JJ et al (2017) Visual exploration of semantic relationships in neural word embeddings. IEEE Trans Vis Comput Graph 99:1–1
10. Zhang L, Chen C (2017) Sentiment classification with convolutional neural networks: an experimental study on a large-scale chinese conversation corpus. In: IEEE international conference on computational intelligence and security, pp 165–169
11. Zhuang H, Wang C, Li C et al (2017) Natural language processing service based on stroke-level convolutional networks for Chinese text classification. In: IEEE international conference on web services, pp 404–411
12. Chen X, Xu L, Liu Z et al (2015) Joint learning of character and word embeddings. In: International conference on artificial intelligence, pp 1236–1242
13. Lai S, Xu L, Liu K, Zhao (2015) Recurrent convolutional neural networks for text classification. In: AAAI, pp 2267–2273
14. Li Y, Wang X, Xu P (2018) Chinese text classification model based on deep learning. Future Internet 10(11):113
15. Yang J, Lyu Q, Gao S et al (2017) Review aspect extraction based on character-enhanced embedding models. In: IEEE international conference on network infrastructure and digital content, pp 219–223
16. Zhang X, Zhao J, Lecun Y (2015) Character-level convolutional networks for text classification. In: Advances in neural information processing systems, pp 649–657
17. Zhou Y, Xu J, Cao J et al (2017) Hybrid attention networks for Chinese short text classification. Computación y Sistemas 21(4):759–769
18. Cho K, Van Merrienboer B, Gulcehre C et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078
19. Bafna P, Pramod D, Vaidya A (2016) Document clustering: TF-IDF approach. In: IEEE international conference on electrical, electronics, and optimization techniques, pp 61–66
20. Qu Z, Song X, Zheng S, Wang X, Song X, Li Z (2018) Improved Bayes method based on TF-IDF feature and grade factor feature for chinese information classification. In: 2018 IEEE international conference on big data and smart computing, pp 677–680
21. Le QV, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning, pp 1188–1196
22. Socher R, Huval B, Manning CD, Ng AY (2012) Semantic compositionality through recursive matrix-vector spaces. In: The 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp 1201–1211
23. Pengfei Liu X, Qiu X, Chen S, Wu XH (2015) Multi-timescale long short-term memory neural network for modelling sentences and documents. In: The 2015 conference on empirical methods in natural language processing, pp 2326–2335
24. Kalchbrenner N, Grefenstette E, Blunsom P (2014) A convolutional neural network for modelling sentences. In: Proceedings of the 52nd ACL, pp 655–665
25. Kim Y (2014) Convolutional neural networks for sentence classification. In: Proceedings of the 2014 conference on EMNLP, pp 1746–1751
26. Yang Z, Yang D, Dyer C et al (2017) Hierarchical attention networks for document classification. In: Conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1480–1489
27. Li Y, Li W, Sun F, Li S (2015) Component-enhanced chinese character embeddings. In: Proceedings of the 2015 conference on EMNLP, pp 829–834
28. Zhou Y, Xu B, Xu J et al (2017) Compositional recurrent neural networks for chinese short text classification. In: IEEE international conference on web intelligence, pp 137–144
29. Gao S, Ramanathan A, Tourassi G (2017) Hierarchical convolutional attention networks for text classification. In: The 3rd workshop on representation learning for NLP, pp 11–23
30. Su J, Zeng J, Xiong D et al (2018) A hierarchy-to-sequence attentional neural machine translation model. IEEE/ACM Trans Audio Speech Lang Process 26(3):623–632
31. Gao L, Guo Z, Zhang H et al (2017) Video captioning with attention-based lstm and semantic consistency. IEEE Trans Multimed 19(9):2045–2055
32. Yang Z, He X, Gao J et al (2016) Stacked attention networks for image question answering. In: IEEE conference on computer vision and pattern recognition, pp 21–29
33. Zhou P, Shi W, Tian J et al (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: Meeting of the association for computational linguistics, pp 207–212
34. Wang L, Cao Z, Melo GD et al (2016) Relation classification via multi-level attention CNNs. In: Meeting of the association for computational linguistics, pp 1298–1307
35. Ling Y, An Y, Liu M et al (2017) Integrating extra knowledge into word embedding models for biomedical NLP tasks. In: IEEE international joint conference on neural networks, pp 968–975
36. Erhan D, Bengio Y, Courville A et al (2010) Why does unsupervised pre-training help deep learning. J Mach Learn Res 11(3):625–660
37. Wang Q, Xu J, Chen H et al (2017) Two improved continuous bag-of-word models. In: IEEE international joint conference on neural networks, pp 2851–2856
38. Wang J, Liu F, Qin S (2017) Global exponential stability of uncertain memristor-based recurrent neural networks with mixed time delays. Int J Mach Learn Cybern 2:1–13
39. Na Liu F, Chen M, Lu (2013) Spectral co-clustering documents and words using fuzzy K-harmonic means. Int J Mach Learn Cybern 4(1):75–83

40. Li P, Yan Ye (2016) Chinese spam filtering based on back-propagation neural networks. Softw Eng 4(2):9–12
41. Sang L, Xie F, Liu X et al (2017) WEFEST: word embedding feature extension for short text classification. In: IEEE international conference on data mining workshops, pp 677–683
42. Musa AB (2013) Comparative study on classification performance between support vector machine and logistic regression. Int J Mach Learn Cybern 4(1):13–24
43. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
44. Zhang D, Wang D (2015) Relation classification via recurrent neural network. Comput Sci. arXiv:1508.01006
45. Graves A, Jürgen Schmidhuber (2005) Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw 18(5):602–610
46. Yong Z, Meng JE, Venkatesan R et al (2016) Sentiment classification using comprehensive attention recurrent models. In: IEEE international joint conference on neural networks, pp 1562–1569
47. Luong MT, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. In: Conference on empirical methods in natural language processing, pp 1412–1421