Annals of Operations Research
https://doi.org/10.1007/s10479-022-04799-w

ORIGINAL RESEARCH

Intelligence customs declaration for cross-border e-commerce based on the multi-modal model and the optimal window mechanism

Xiaofeng Li[1] · **Jing Ma**[1] · **Shan Li**[1]

Accepted: 23 May 2022

Abstract
This paper aims to study the intelligent customs declaration of cross-border e-commerce commodities from algorithm design and implementation. The difficulty of this issue is the recognition of commodity names, materials, and processing processes. Because the process of recognizing these three kinds of commodity information is similar, this paper chooses to identify the commodity name as the experimental research object. The algorithm in this paper is based on the premise of pre-clustering, using an optimal window mechanism to obtain the best word embedding vector representation. The Vision Transformer model extracts image features instead of traditional CNN models, and then text features are fused with image features to generate a multi-modal semantically feature vector. Finally, a deep forest classifier replaces the conventional neural network classifiers to complete the commodity name recognition task. The experimental results show that, for more than 600 different commodities on the 120,000 data records, the precision is 0.85, the recall is 0.87, and the F_1_score is 0.86. So, our algorithm can effectively and accurately recognize e-commerce commodity names and provide a new perspective on the research of e-commerce intelligence declarations.

Keywords Cross-border electronic commerce · Commodity name recognition · Optimal window mechanism · Vision transformer · Multi-modal

✉ Xiaofeng Li
lxf0895@nuaa.edu.cn

Jing Ma
majing5525@126.com

Shan Li
lishan@nuaa.edu.cn

[1] College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, Jiangsu, China

Published online: 15 June 2022

1 Introduction

Since 2020, the global e-commerce industry has withstood the severe test of the COVID-19 epidemic and has shown strong resilience. Cross-border e-commerce has become a significant force to ensure global consumer demand for goods in the wake of the epidemic. China's cross-border e-commerce industry has also grown by leaps and bounds. According to data from the General Administration of Customs of China, the total import and export volume of cross-border e-commerce in China increased by 31.1% to 1.69 trillion yuan in 2020. The number of cross-border e-commerce declarations submitted to China Customs for the year was 2.45 billion, up 63.3% percent year-on-year (CSY, 2020). The development of e-commerce and cross-border e-commerce in China is shown in Figs. 1 and 2.

With cross-border e-commerce rapid development, there has been a surge in the number of transaction goods and an exponential increase in declarations. The artificial mode is the current mode that declaration systems mainly use. When the number of declared commodities increases dramatically, it will be inefficient. In addition, the declaring process of goods requires the filling of three crucial information: the name of the item, material, and processing process. This information is used to code the commodity uniquely. The code is named the international HSCode (International Convention for Harmonized Commodity Description and Coding System). Each HSCode corresponds to the tax rate, so if the commodity information is incorrect, the declaration may fail, resulting in a lot of time and cost losses.

In addition to the inefficiency of the manual mode, there are also shortcomings such as high cost, high error rate, and experience dependence. We attempt to use artificial intelligence algorithms to replace the current manual declaration process to achieve automatic identification of commodity information, automatic declaration, which will significantly improve the accuracy and efficiency of the declaration.

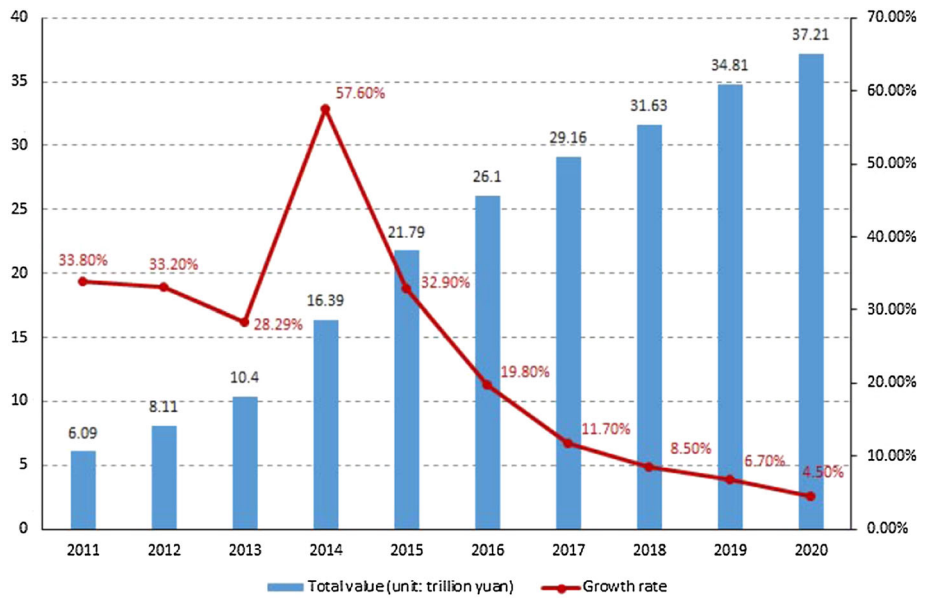

Fig. 1 The development of Chinese e-commerce

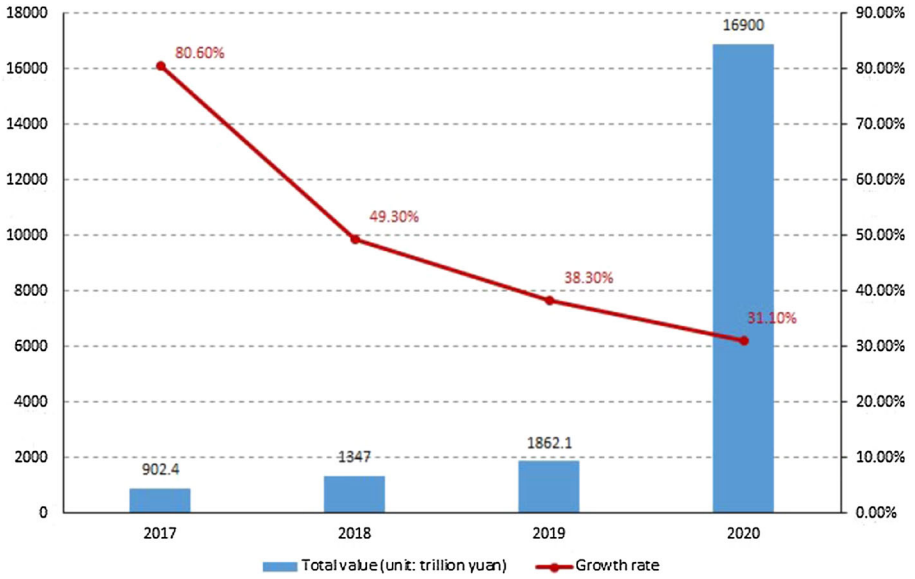

Fig. 2 The development of Chinese cross-border e-commerce

To achieve intelligent recognition of commodity information is to design algorithmic models to identify commodity names, material, and processing processes separately in the short text of commodity description. Since the process of recognizing these three types of information mentioned above is similar, we choose to identify the name of a commodity as the primary research issue of this thesis.

After studying the commodity description short texts dataset, we found that the commodity text describes the commodity characteristics. It usually contains many weak-correlation words to maximize the likelihood of this commodity being gotten by search engines. For example, the description text is "Galaxy, Guardian, Treeman, Groot, Cartoon, Ceramics, Mug, Animation, Movies, Periphery, Marvel, Water Cup". Although there are a lot of words in this short text, the standard commodity name should be "Mug". While plenty of other words are around "Mug", and these words are weak-correlated to each other, and some of them even have no semantic relationship with the standard commodity name ("Mug").

Such texts are characterized by random co-occurrence of words, weak correlations between words, colloquial semantic expressions, and repeated synonyms. Although this kind of expression can improve the hit rate of search engines, it also increases the difficulty of recognizing the commodity name.

This paper proposes a novel algorithm based on multi-modal fusion and the optimal window mechanism to address this real-world issue. The algorithm will process numerous cross-border e-commerce commodity description texts and images and recognize the specification commodity name. Meanwhile, this algorithm will provide some technical support for intelligent customs declarations.

The research works presented here advance research in short text classification by constructing the WWV-gcForest (Word2Vec-Window optimization-ViT-gcForest) model. The contributions of this article can be summarized as follows:

(1) In this paper, we innovatively propose an optimal window mechanism suitable for the Word2Vec model to improve the quality of short-text word embedding significantly.

(2) In this paper, we compare the word embedding effect of the Bert model, the Word2Vec model, and the Word2Vec model with the optimal window mechanism on the data set by a visualization method.
(3) In this paper, the Vision Transformer model is used instead of classical CNN models to complete the extraction of commodity image features, which improves the quality of image features extraction.
(4) The deep forest model is used to replace the neural network models to complete the name recognition task, which provides a new research perspective for the cross-border e-commerce name recognition issue.

2 Related work

As far as we know, some related research works about this issue were started in 2019. Initially, some researchers classified this issue as named entity recognition (NER). Luo et al. (2020) used Word2Vec vectors fused with TF-IDF vectors to represent text features and then used the LSTM model to implement commodity entity recognition. The accuracy was about 70%. Subsequently, Ma et al. (2019) optimized the Word2Vec model, the TF-IDF model, and the SVM model, improving commodity name recognition accuracy. Although these algorithms have achieved good results, such algorithms only use text features and ignore the role of commodity image features. Image features and text features fusion algorithms have been used to solve these issues. For example, Zhu et al. (2020) extracted both image and textual features to evaluate e-commerce product values. At the same time, Chen et al. (2021) used the Transformer model to classify e-commerce products, and this research work is beneficial for our study.

From our point of view, how to recognize a commodity name from a commodity description text can be abstracted into a short text multi-classification issue. Each type of commodity name constitutes an independent category, and different commodity names belong to different categories. Our research aims to classify each commodity description text into its category accurately. Therefore, the research approach can also use short text multi-classification algorithms.

Short text classification algorithms were developed along with text classification algorithms, while the short-text classification issue is more complex than the long-text classification issue. Because short texts usually contain less information, classifier models can only learn fewer features. Before the 1980s, text classification algorithms were mainly based on rule-matching algorithms. Those algorithms were highly dependent on prior experience and had inherent shortcomings such as poor generalization and robustness.

The development of machine learning has provided many classical algorithms for short text classification, such as the SVM algorithm proposed by Cortes and Vapnik (1995) and the Random Forest algorithm proposed by Breiman (2001), and so on. Alfaro et al. (2016) combined SVM with KNN to propose a multi-stage text classification algorithm for weblog short text classification with remarkable results. Meanwhile, the CRF algorithm allows us to classify the current short text according to the labeled contents and the contents recognized by the algorithm. For example, Kumar et al. (2018) used the CRF model to achieve encoding and classification short-texts of dialogues. So, it contributes significantly to improving the accuracy of short text classification. In addition, to better solve the Chinese short-text classification issue, Sun and Chen (2018) extended the Chinese short text features by using the LDA model to improve the classification effect of Chinese short text.

With the rise of deep learning, short text classification algorithms based on neural networks have sprung up in recent years. Initially, Pennington et al. (2014) proposed the GloVe model, and Mikolov et al. (2013) proposed the Word2Vec model, which provided a shallow neural network-based feature vector representation for vectorization of short texts. Then, composite classification models based on these vector models have been proposed one after another, such as the FastText model proposed by Joulin et al. (2017), the TextCNN model proposed by Chen (2015), the multi-channel TextCNN model proposed by Guo et al. (2019) and so on. These models were used with shallow word embedding models and achieved better results, opening up new perspectives for solving short text classification issues.

However, shallow neural network models have an inherent deficiency in contextual semantic understanding. Static word vector representation cannot solve problems such as multiple meanings of a word. Deep neural network models such as the LSTM model (Hochreiter & Schmidhuber, 1997) and the GRU model (Cho et al., 2014) have emerged gradually to solve the problem of extracting contextual semantic features of texts. At the same time, the BERT (Kenton & Toutanova, 2019) language model also stood up for dynamic word vector representation. The most important contribution of the BERT model is the mask mechanism, which can more fully obtain contextual semantic information and thus can better solve the problem of multiple meanings of a word.

The Bert language model is based on a pre-training mechanism, and it can also extract contextual semantic information from texts. However, it requires more training data and mighty computing power in its pre-training and fine-tuning process. The performance of Bert models is not very satisfactory in some small-scale datasets or some specific domains (Burdick et al., 2021). In addition, many commodity images should also be fully utilized. Thus, Kumar et al. (2020) used the VGG16 model to extract image features and classify Twitter short texts. Using image features provides a new idea for studying short-text classification issues. Therefore, this paper proposes a novel algorithm that fuses text features with image features to generate multi-modal feature representations and employs an optimal window mechanism, which significantly improves the classification accuracy of short texts and achieves commodity name recognition.

3 Research methodology

3.1 Data acquisition

The dataset that we used to train and evaluate the WWV-gcForest model is crawled from the Tmall e-commerce platform. After filtering data and data augmentation, the dataset has about 120,000 short texts for more than 600 different commodities. These commodities represent the most popular commodity categories during "the Nov 11 online shopping carnival". Therefore, if we can accurately identify the names of these commodities, the declaration accuracy will be improved, and labor and time costs will also be significantly reduced.

The corpus has been segmented using a user dictionary and the jieba system and removed some stop words. A part of it is translated into English for display purposes, as shown in Fig. 3.

3.2 Data augmentation and balance

In our dataset, there is a problem of imbalanced amounts of various commodity description texts. Some commodities have abundant description texts, while others have very few. Unbal-

	A	B
1	**Commodity Description**	**Lable**
2	Zone D, USA, new fashion, print, sleeveless, slim fit, **dress**	dress
3	AV to VGA, **converter**, AV line, VGA line, set-top box, monitor, watch TV, monitor, watch TV	converter
4	Korean version, simple, solid color, student, bottoming shirt, round neck, loose, cotton short sleeve, **T-shirt**	T-shirt
5	Special offer, summer, short-sleeved shirt, men's wear, Korean version, thin section, non-iron, slim, white, **shirt**, men, casual	shirt
6	France, Sherpa, mountaineering glasses, alpine, **glasses**, stock	glasses
7	Long distance, airplane, travel, portable, inflatable, pillow, neck protection,**U-shaped pillow**, cervical pillow, sleeping, artifact	U-shaped pillow
8	Spring, men, casual, slim, oxford, solid color, bottoming shirt, student, short sleeve,Korean version, white, **shirt**	shirt
9	White, cotton, Korean version, waist, slim, large size women's clothing, spring and summer women's clothing, new, hollow, **dress**	dress
10	V-neck, retro, print, loose, ladies, chiffon, **dress**, ethnic style, women's wear, elegant, long skirt, new	dress
11	**LED light**, flashing lights, string lights, curtain lights, wedding rooms, furnishings, window, decoration, storefront, courtyard, stars, curtains, backlights, **LED light**	LED light
12	HDMI HD cable,adapter, Apple, PC, TV, projector, **converter**	converter

Fig. 3 A part of the corpus (Raw Data)

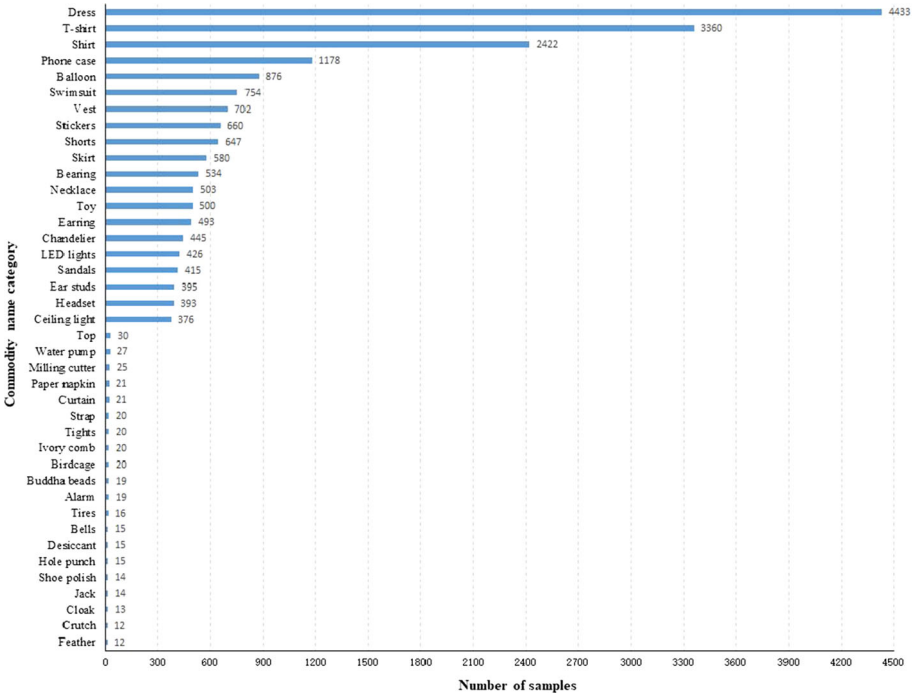

Fig. 4 The number of category samples (before data balancing process)

anced samples of commodity description texts will cause over-fitting of the model, which will seriously affect the model's accuracy. So, it is necessary to filter the acquired raw data and delete some such as incorrect, duplicate, invalid, etc. Then, we evaluated the number of category samples on the filtered dataset. The 20 commodities with the largest sample size and the 20 commodities with the smallest sample size were selected, respectively, shown in Fig. 4.

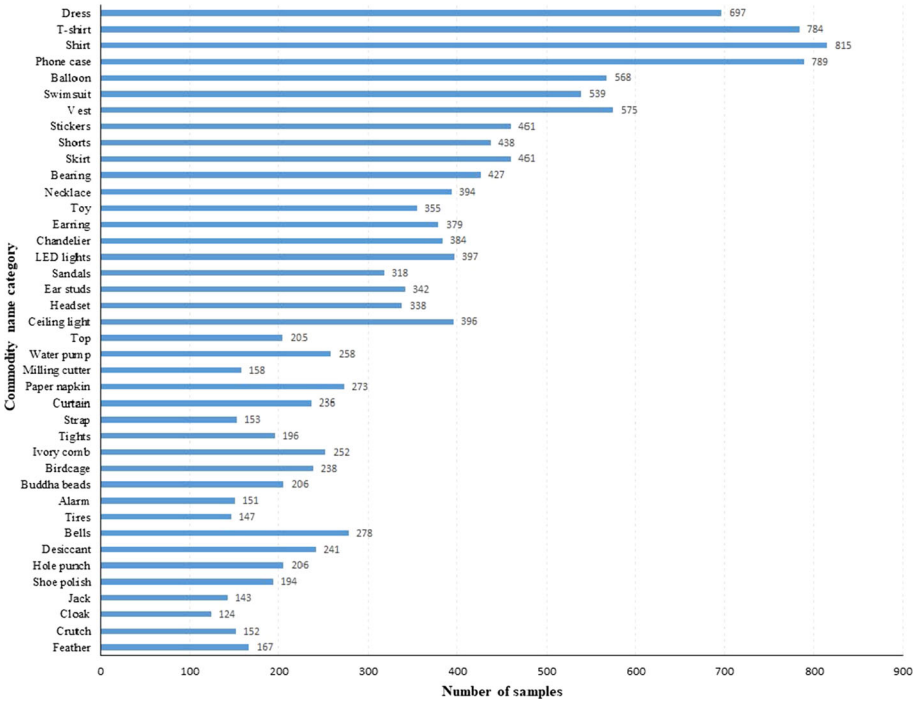

Fig. 5 The number of category samples (after data balancing process)

The random down-sampling process is used for those commodity texts with more than 300 samples. While those commodity texts with less than 100 samples, data augmentation approaches are used to increase the original number of samples randomly. Synonym Replacement, Random Insertion, Random Swap, and Random Deletion are used to augment samples. After the data balancing process, the distribution of the samples number is shown in Fig. 5.

Compared to Fig. 4, the number of samples shown in Fig. 5 is significantly more balanced. Through the data balancing process, the more balanced dataset has contributed substantially to improving the accuracy of models. The specific evaluation of its effect can be found in the experiment section.

3.3 The Word2Vec text feature extraction model

The Word2Vec model is a probabilistic language model based on the neural network proposed by Mikolov et al. (2013), which provides a method to represent text features based on distributed multi-dimensional vectors. The word vectors are based on the co-occurrence and internal semantic information of words in existing contexts, calculated by a neural network. The Word2Vec model includes the CBOW training module and the Skip-gram training module. The CBOW (continuous bag of words) model is used in our experiments. The principle of the CBOW model is to use contextual content to predict the current word. The algorithm predicts the target word $\boldsymbol{W_t}$ under the condition that $\boldsymbol{W_{t-2}}$, $\boldsymbol{W_{t-1}}$, $\boldsymbol{W_{t+1}}$, and $\boldsymbol{W_{t+2}}$ are known. The model consists of an input layer, a projection layer, and an output layer. The specific network structure is shown in Fig. 6.

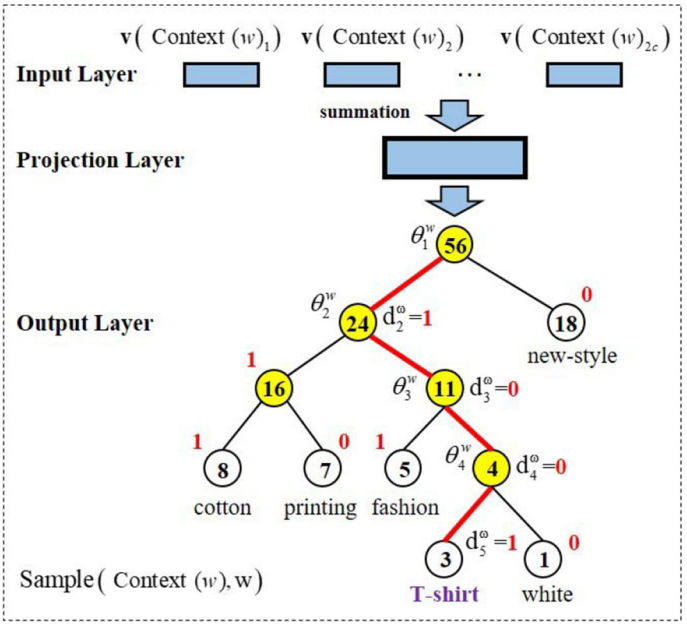

Fig. 6 The network structure of the CBOW module

Input layer A word vector $\boldsymbol{v}(\text{ Context }(\boldsymbol{w})_{\mathbf{1}})$ containing 2c words in $\boldsymbol{v}(\text{ Context }(\boldsymbol{w})_{\mathbf{1}})$, $\boldsymbol{v}(\text{ Context }(\boldsymbol{w})_{\mathbf{2}})$, . . . , $\boldsymbol{v}(\text{ Context }(\boldsymbol{w})_{\boldsymbol{2c}}) \in \mathbf{R_m}$, m represents a word vector length, c indicates that c words are taken before and after the word w.

Projection layer The 2c vectors are summed and accumulated as shown below (Eq. 1).

$$X_{\mathrm{w}} = \sum_{i=1}^{2c} V\ (\ Context\ (w)_i) \in R^m \tag{1}$$

Output layer The output layer is a Huffman tree with the corpus words as the leaf node and the word frequency as the weight. The number of leaf nodes of the tree is $\boldsymbol{N}(\boldsymbol{N} = \mid \boldsymbol{D} \mid)$, the number of non-leaf nodes is $\mathbf{N} - \mathbf{1}$.

This paper uses the CBOW module to train the word vector and the log-likelihood function as the objective function (Eq. 2).

$$\mathrm{L} = \sum_{w \in \mathrm{C}} \log P(w \mid Context(w)) \tag{2}$$

The Huffman Tree coding and the Hierarchical Softmax are used in the CBOM module. The Huffman tree is the tree with the shortest path length with weights, the nodes with larger weights are closer to the root, and the weights of the left leaf nodes are larger than the weights of the right leaf nodes in the same layer, and the path of the left node is encoded as '1', and the path of the right node is encoded as '0'.

As shown in Fig. 6 **w** = 'T-shirt', from the root node to the word 'T-shirt', there are four branches: the four red lines, and for this Huffman tree, each branch is equivalent to a binary category. The Word2Vec model defines a negative class as '1' and a positive class as '0', which means that if it is divided into a left subtree, it is a negative class; if it is divided into

a right subtree, it is a positive class. So, the node path from the root node to the aim node (**w** = T-shirt) in Fig. 6 is coded as '1001'.

It is necessary to introduce the objective function of the CBOM module in detail, and it is closely related to the optimal window mechanism. To present the derivation of the objective function, we defined the following variables.

① $\boldsymbol{p^w}$: The path is from the root node to the target node.

② $\boldsymbol{l^w}$: Number of nodes contained in the path $\boldsymbol{p^w}$.

③ $\boldsymbol{p_1^w}, \boldsymbol{p_2^w}, \ldots, \boldsymbol{p_{lw}^w}$: The nodes in the path $\boldsymbol{p^w}$,where $\boldsymbol{p_1^w}$ is the root node and $\boldsymbol{p_{lw}^w}$ is the node where the word is located.

④ $\boldsymbol{d_2^w}, \boldsymbol{d_3^w}, \ldots, \boldsymbol{d_{lw}^w} \in \{0, 1\}$: Huffman encoding of word w. The Huffman encoding of a word is composed of $\left(\boldsymbol{l^w - 1}\right)$ bits. $\boldsymbol{d_j^w}$ denotes the Huffman encoding of the jth word in the path $\boldsymbol{p^w}$, and the root node is not involved in the encoding.

⑤ $\boldsymbol{\theta_1^w}, \boldsymbol{\theta_2^w}, \ldots, \boldsymbol{\theta_{l^{w-1}}^w} \in \mathbf{R^m}$:The vector of non-leaf nodes in the path $\boldsymbol{p^w}$ and $\boldsymbol{\theta_j^w}$ denotes the vector of the jth non-leaf node in the path $\boldsymbol{p^w}$.

The Sigmoid function is used in the calculation of binary classification. The probability of a node being classified as a positive class is shown in Eq. 3. In contrast, the probability of being classified as a negative class is shown in Eq. 4. $\boldsymbol{\theta}$ is the non-leaf vector $\boldsymbol{\theta_j^w}$.

$$\sigma\left(x_w^T\theta\right) = \frac{1}{1 + e^{-x_w^t\theta}} \tag{3}$$

$$1 - \sigma\left(x_n^T\theta\right) \tag{4}$$

As shown in Fig. 6, the probability of reaching the leaf node "T-shirt" from the root node is expressed as Eqs. 5, 6, 7, and 8.

The first time:

$$p\left(d_2^w \mid x_w, \theta_1^w\right) = 1 - \sigma\left(x_w^T\theta_1^w\right) \tag{5}$$

The second time:

$$p\left(d_3^w \mid x_w, \theta_2^w\right) = \sigma\left(x_w^T\theta_2^w\right) \tag{6}$$

The third time:

$$p\left(d_4^w \mid x_w, \theta_3^w\right) = \sigma\left(x_w^T\theta_3^w\right) \tag{7}$$

The fourth time:

$$p\left(d_5^w \mid x_w, \theta_4^w\right) = 1 - \sigma\left(x_w^T\theta_4^w\right) \tag{8}$$

The probability of the word **w** = T-shirt is calculated as shown in Eq. 9. The general representation of this conditional probability is shown in Eq. 10.

$$p(\text{ T-shirt} \mid \text{Context(T-shirt)}) = \prod_{j=2}^{5} p\left(d_j^w \mid x_w, \theta_{j-1}^w\right) \tag{9}$$

$$p(w \mid \text{Context}(w)) = \prod_{j=2}^{l^w} p\left(d_j^w \mid x_w, \theta_{j-1}^w\right) \tag{10}$$

Here

$$p\left(d_j^w \mid x_w, \theta_{j-1}^w\right) = \begin{cases} \sigma\left(x_w^T\theta_{j-1}^w\right), & d_j^w = 0 \\ 1 - \sigma\left(x_w^T\theta_j^w - 1\right), & d_j^w = 1 \end{cases} \tag{11}$$

Substituting (10) into (2) yields

$$\begin{aligned} L &= \sum_{w \in C} \log \prod_{j=2}^{l^w} \left[\sigma\left(x_w^T \theta_{j-1}^w\right)\right]^{1-d_j^w} \cdot \left[1-\sigma\left(x_w^T \theta_{j-1}^w\right)\right]^{d_j^w} \\ &= \sum_{w \in C} \sum_{j=2}^{l^w} \left(1-d_j^w\right) \cdot \log\left[\sigma\left(x_w^T \theta_{j-1}^w\right)\right] + d_j^w \cdot \log\left[1-\sigma\left(x_w^T \theta_{j-1}^w\right)\right] \end{aligned} \tag{12}$$

For the convenience of calculation, let

$$L(w, j) = \left(1-d_j^w\right) \cdot \log\left[\sigma\left(x_w^T \theta_{j-1}^w\right)\right] + d_j^w \cdot \log\left[1-\sigma\left(x_w^T \theta_{j-1}^w\right)\right] \tag{13}$$

Then, we have

$$\begin{aligned} \frac{\Delta L(w, j)}{\Delta \theta_{j-1}^w} &= \left\{\left(1-d_j^w\right)\left[1-\sigma\left(x_w^T \theta_{j-1}^w\right)\right] x_w - d_j^w \sigma\left(x_w^T \theta_{j-1}^w\right)\right\} x_w \\ &= \left[1-d_j^w-\sigma\left(x_w^T \theta_{j-1}^w\right)\right] x_w \end{aligned} \tag{14}$$

Thus, the update equation of $\boldsymbol{\theta_{j-1}^w}$ is

$$\theta_{j-1}^w = \theta_{j-1}^w + \eta\left[1-d_j^w-\sigma\left(x_w^T \theta_{j-1}^w\right)\right] x_w \tag{15}$$

Where, $\boldsymbol{\eta}$ is learning rate.

Next, the gradient of with $\mathbf{L(w, j)}$ respect to $\boldsymbol{x_w}$ is

$$\frac{\Delta L(w, j)}{\Delta x_w} = \left[1-d_j^w-\sigma\left(x_w^T \theta_{j-1}^w\right)\right] \theta_{j-1}^w \tag{16}$$

The gradient accumulation of $\boldsymbol{x_w}$ is used to update $\boldsymbol{v(\tilde{w})}$ and obtain the vector representation of the word $\mathbf{w}$.

$$v(\tilde{w}) = v(\tilde{w}) + \eta \sum_{j=2}^{l^w} \frac{\Delta L(w, j)}{\Delta x_w}, \tilde{w} \in \text{Context}(w) \tag{17}$$

3.4 The optimal window mechanism

After reading a lot of related papers, we found that the sliding window value is usually [5,10]. We innovatively proposed to enlarge the window value search range under the premise of data pre-clustering to find the optimal window value suitable for the current data set. We presented the optimal window mechanism based on theoretical analysis and experimental validation.

Short texts usually contain less text and, therefore, less information in the text. The purpose of searching for the optimal window on the pre-clustered dataset is to expand the current short text content using similar commodity texts in the same category to increase information and improve the accuracy of feature extraction. The optimal window mechanism proposed in this paper is described as follows.

Firstly, the short texts of commodity descriptions are pre-clustered by labels. Then a grid search strategy is used to increase the sliding window size in turn and verify the effect of the current window size by the classification model precision. The optimal window for this

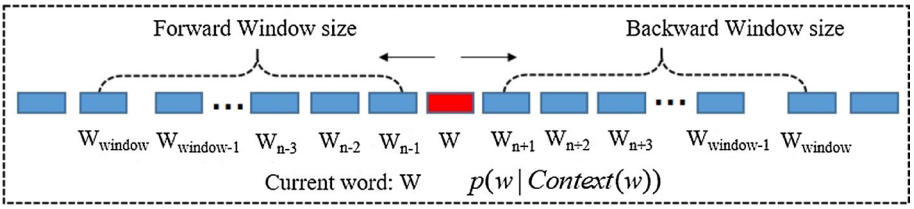

Fig. 7 The window size diagram

dataset is selected based on the highest value of classification precision. The forward and backward sliding windows for the current word W are shown in Fig. 7.

We validate the effectiveness of the optimal window mechanism on multiple types of datasets, such as short texts, long texts, Chinese texts, English texts, and so on. We need to note that the optimal window value is closely related to the dataset, and the optimal window size is set differently for different types of datasets. We will give the theoretical analysis about it in the experimental section.

3.5 The ViT image feature extraction model

The ViT (Vision Transformer) model is mainly used to accomplish image classification, target identification, and image segmentation tasks. It borrowed the transformer model from the natural language processing domain. The transformer model was innovatively migrated to the computer vision domain by Dosovitskiy et al. (2020).

The core idea of the ViT model is to divide an image into fixed-size patches and then obtain the embedding representation of the patches by a linear transformation, which is similar to the process of word segmentation and word embedding. The ViT model only uses the Encoder part of the transformer model to extract the image features and adopts two embedding mechanisms, Patch Embedding and Position Embedding. The input format of the transformer model is a sequence of word embeddings, so the patches of the image can be divided into a fixed order and input to the transformer model. Then the image features will be extracted. The process of ViT model for extracting image features in this paper is shown in Fig. 8.

Patch Embedding is the process of converting the current two-dimensional image into a set of ordered one-dimensional patches. Define the current two-dimensional image as $\boldsymbol{x} \in \boldsymbol{R}^{\boldsymbol{H}\times\boldsymbol{W}\times\boldsymbol{C}}$,where $\mathbf{H}$ and $\mathbf{W}$ mean the height and width of the image, respectively, and C means the number of image channels. For example, if the current image is in RGB mode, $\mathbf{C} = \mathbf{3}$. The current image was divided into patches with $\mathbf{P}^*\mathbf{P}$ size. Then, these patches were reshaped into a set of ordered patches $\boldsymbol{x} \in \boldsymbol{R}^{\boldsymbol{H}\times\boldsymbol{W}\times\boldsymbol{C}}$.The current image is sliced into $\boldsymbol{N} = \boldsymbol{H} * \boldsymbol{W}/\boldsymbol{P}^{\boldsymbol{2}}$ patches whose composed sequence length is $\boldsymbol{P}^{\boldsymbol{2}} \cdot \boldsymbol{C}$. Then, by a linear transformation, the patches are mapped to the dimension of the user's desired size, and the embedding process of the patches is completed. This paper uses RGB pattern images, i.e., $\mathbf{C} = \mathbf{3}$. The image size is 224*224, i.e., $\mathbf{H} = \mathbf{W} = \mathbf{224}$, the size of the generated patches is 32*32, i.e., $\mathbf{P} = \mathbf{32}$, each image generates 49 patches. The length of the sequence generated after linear transformation is 32*32*3 = 3072. Finally, it is linearly mapped to a 128-dimensional vector. The patches are flattened and mapped to 128 dimensions with a trainable linear projection (Eq. 18). The output of this projection is called patch embedding.

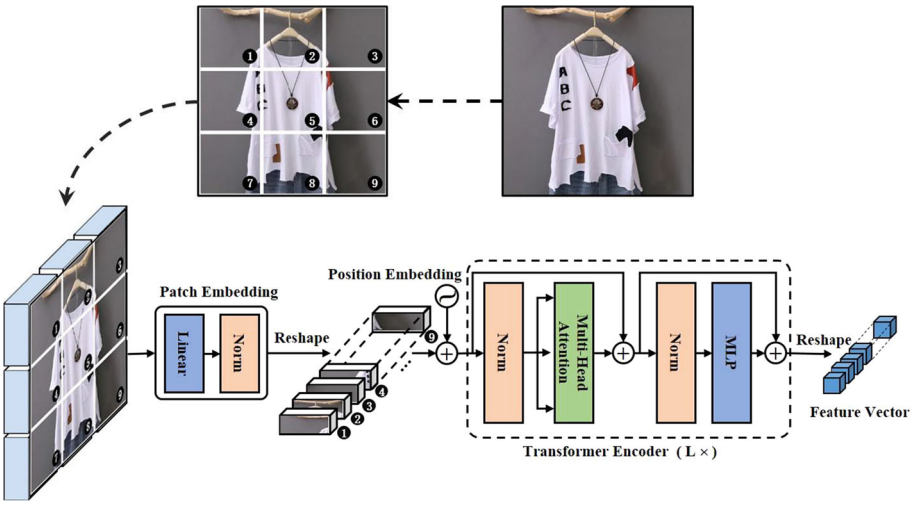

Fig. 8 The ViT model extracts image feature process

Position embedding The model also needs to obtain the position information of the patches. The position of the patches is used to encode the token information and calculate Self-Attention.

The Transformer encoder consists of alternating layers of multi-head self-attention (Wang et al., 2017) (MSA, Eq. 19) and MLP (Eq. 20) blocks. Layer-norm (LN, Eq. 21) is applied before every block and residual connections after every block. The MLP contains two layers with a GELU non-linearity.

$$\mathbf{z}_0 = \left[\mathbf{x}_{\text{class}}\,;\, \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots ; \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{\text{pos}}\,,$$
$$\mathbf{E} \in \mathrm{R}^{(P^2\cdot C)\times D}, \mathbf{E}_{\text{pos}} \in \mathrm{R}^{(N+1)\times D} \tag{18}$$
$$\mathbf{z}'_\ell = \text{MSA}\left(\text{LN}\left(\mathbf{z}_{\ell-1}\right)\right) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \ldots L \tag{19}$$
$$\mathbf{z}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{z}'_\ell\right)\right) + \mathbf{z}'_\ell, \quad \ell = 1 \ldots L \tag{20}$$
$$\mathbf{y} = \text{LN}\left(\mathbf{z}_L^0\right) \tag{21}$$

where $\boldsymbol{Z_0}$ is the current transformer layer output, $\boldsymbol{Z_{\ell-1}}$ is the previous transformer output, $\boldsymbol{X}_{class}$ is the classification information of the current image, $\boldsymbol{E}_{pos}$ is the position information, $\boldsymbol{E}$ is the linear transformation factor, $\boldsymbol{N}$ is the number of patches, **P*P** is the size of a patch, $\boldsymbol{D}$ is the number of dimensions.

The role of the activation function is to give the network model a nonlinear fitting capability. GELU (Gaussian Error Linear Unit) is a high-performance neural network activation function. The formula is

$$\text{GELU}(x) = xP(X \le x) = x\Phi(x) \tag{22}$$

where, $\boldsymbol{\Phi(x)}$ is the probability function of the **x** Gaussian distribution. The complete representation of $\boldsymbol{\Phi(x)}$ is

$$\text{xP}(\text{X} \le \text{x}) = x\int_{-\infty}^{x} \frac{e^{-\frac{(X-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}\text{d}X \tag{23}$$

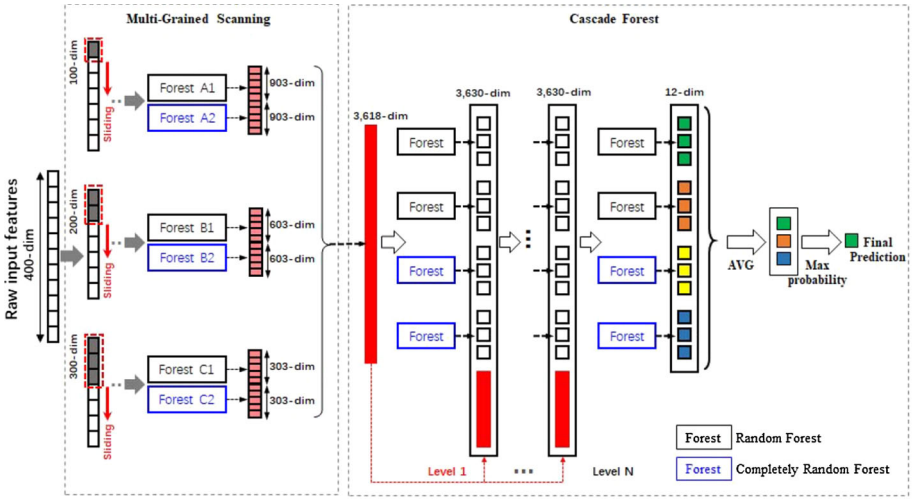

Fig. 9 The gcForest model structure

Here, $X \sim N(\mu, \sigma^2)$, μ is the mathematical expectation, σ^2 is the variance. If $X \sim N(0, 1)$ (μ=0, σ^2=1), we can approximate the GELU with

$$\mathrm{GELU}(x) = 0.5x\left(1 + \tanh\left[\sqrt{2/\pi}\left(x + 0.044715x^3\right)\right]\right) \tag{24}$$

3.6 The gc-Forest classifier model

The gcForest (Multi-Grained Cascade Forest) is a deep tree integration method that enables multi-batch integration of decision trees through a cascade structure between random forests (Zhou & Feng, 2019). The model defines two types of random forests, completely random tree forests, and random forests. A completely random tree forest contains 1000 completely random trees. The division of parent nodes in a completely random tree is achieved by randomly selecting features among the data features. The condition for a completely random tree to stop growing is that the number of leaf nodes of the same type in the current layer is greater than 10, or the number of sample classifications of leaf nodes is greater than 10.

Each forest in a random forest also contains 1000 completely random trees. While the most significant difference between a random forest and a completely random forest is the division of the parent node. It is a random selection of $\sqrt{n}$ features among the total number of current features n. Among these $\sqrt{n}$ features selected, some more features are selected by Gini coefficients for splitting.

Each decision tree outputs probability values for each class in the sample classification process. After all the decision trees in the random forest have output probability values, the final probability vector for each class is obtained by averaging the vectors for each class output by all the decision trees. The output vectors of the forest at the same level are concatenated to generate a new vector, and this new vector is used as the input vector for the next level. The vector with the highest category probability value in the average of the output vectors of the last forest level is used to determine which class the current sample should be classified. The structure of the gcForest model is shown in Fig. 9.

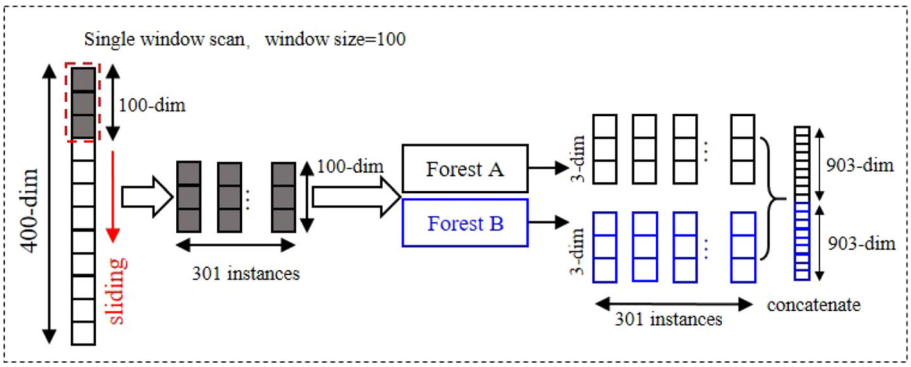

Fig. 10 Single sliding window scanning structure

The gcForest model consists of two main modules: Multi-Grained scanning and cascade forest. Multi-Grained scanning transforms, amplifies, filters, and concatenates the original features. The results are finally transformed into a category probability vector and used as the subsequent input vector. The specific process is described as follows:

(1) Assuming that the sample feature dimension is n, the input sample is sampled with a sliding window of length m to obtain $\boldsymbol{T = (n - m) + 1}$ m-dimensional sub-feature vectors. This process is similar to the sliding operation of a convolution kernel with step size m in a convolutional neural network.
(2) Each subsample will be used in the training of the cascade forests, and each forest will output a probability vector of dimension **x** (x represents the number of categories). After scanning each forest, a **T*x** representation vector will be generated. Finally, the results of the **k** forests of the current layer will be concatenated together as the output vector of the current layer. The Multi-Grained refers to the size of the sliding window m. A simple case of a single sliding window is shown in Fig. 10 (**x = 3**, **m = 100**).

Cascade forest inputs the combined features after Multi-Grained scanning into a hierarchical structure consisting of multiple forests, and each forest will output a new category probability vector. These vectors are concatenated as input vectors for the next layer, and then the final category probability vector is obtained by learning from the multi-grained forest structure.

4 Experiment

4.1 Experimental process design

This experiment is based on the data processing process, and the practical steps are as follows.

(1) In the first step, commodity description short texts are pre-processed, such as data filtering, data balancing, and labeling. Then, the commodity names in the labels are added to the user dictionary, which improves the word segmentation effect. The short text is segmented into words using the jieba word segmentation system and the user dictionary. User stop words are defined and added to the stop word list, and then all stop words in short texts are deleted.

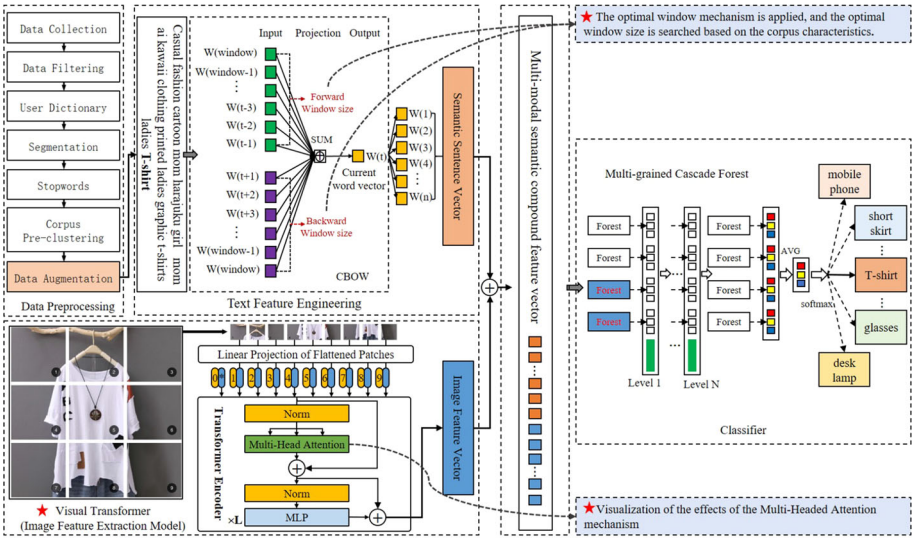

Fig. 11 The cross-border e-commerce name recognition algorithm based on the multi-modal model and the optimal window mechanism

(2) In the second step, the gcForest model is used as a classifier. The optimal window mechanism is applied to search for the window size and visualize the classification precision.
(3) In the third step, text features are extracted, respectively, using the BERT model, the Word2Vec model, and the Word2Vec model with the optimal window mechanism. The classification accuracies of the three models are visualized.
(4) In the fourth step, the vectors represented by the BERT model, the Word2Vec model, and the Word2Vec model with the optimal window mechanism are visualized, respectively, to verify the effectiveness and superiority of the optimal window mechanism.
(5) In the fifth step, the commodity image is divided into a sequence of patches by the ViT model. Then the weights of the patches are assigned using the multi-headed attention mechanism, and then the current image is encoded, which will output a 128-dimensional image feature vector. Meanwhile, the weights assigned by the attention mechanism will be visualized and used to verify the effectiveness of the multi-headed attention mechanism.
(6) In the sixth step, a multi-modal feature vector is generated by concatenating the text feature vector with the image feature vector. It will be used to train the multi-grained cascade forest model. At the same time, the rationality of the selected classifier will be verified. The comparison experiments will be done for various classical classifiers such as SVM, Softmax, XGBoost, etc.
(7) In the seventh step, the multi-grained cascade forest classifier recognized commodity names in the current commodity text based on the maximum probability value.

The dataset is divided into training and test sets in the ratio of 9:1. A data validation method using ten-fold cross-validation will be used. Multiple composite models will be selected for comparison experiments to verify the superiority of our model results. Model ablation experiments will also be done to validate the role of each component of our model. The flow of the cross-border e-commerce name recognition algorithm based on the multi-modal model and the optimal window mechanism is shown in Fig. 11.

4.2 Parameter setting

The following tables list some of the parameter settings for the Word2Vec model, the ViT-Base model, and the gcForest model. The default values are used for parameters not listed in the Tables 1, 2, and 3.

4.3 Experimental evaluation

The three metrics of **Precision**, **Recall**, and $\boldsymbol{F_1}$**_score**, which are widely used in supervised machine learning, are used to evaluate the quality of experimental results.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{25}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{26}$$

$$\text{F}_1\_\text{score} = \frac{2 \cdot \text{Pr} \cdot \text{Re}}{\text{Pr} + \text{Re}} \tag{27}$$

where **TP** is True Positive, **FP** is False Positive, and **FN** is False Negative.

Table 1 Parameter settings for the Word2vec model

Parameter name	Parameter value	Parameter description
Sg	0	Sg = 0, Training with CBOW model
Size	300	The dimension of a feature vector
Window	***600***	***Window size (commonly 5–10)***

The bold italics values indicate that the optimal value of the window is 600 (highlight)

Table 2 Parameter settings for the ViT-Base model

Parameter name	Parameter value	Parameter description
Layers	12	Number of ViT modellayers
hidden_size	768	Number of hidden layer cells
mlp_size	3072	Number of multilayer perceptron
Heads	12	The head number of MHA
image_size	224	Size of the image
patch_size	32	Size of each patch
Channels	3	Channels of the picture pattern
Dim	128	Dimensions of the feature vector
num_patches	$(\text{image\_size}/\text{patch\_size})^2$	How many patches an image divided
patch_dim	$(\text{channels}/\text{patch\_size})^2$	The length of the sequence generated
num_epochs	50	Number of training iterations
train_batch_size	32	Size of each batch during training
lr	1.0e−5	Learning rate
act_layer	GELU	Activation function
Optimizer	Adam	Optimizer selection

Table 3 Parameter settings for the gcForest model

Parameter name	Parameter value	Parameter description
shape_1X	[1,300]	The shape of a single sample element [n_lines, n_cols]
n_mgsRFtree	10	Number of trees in random forests
Window	[3,6]	List of window sizes used during multi-grained scans, sliding with 3, 4, 5, 6, respectively
Tolerance	0.3	The difference in precision of cascading growth, if there is no significant performance gain, the training process will terminate
min_samples_mgs	10	The minimum number of samples to perform splitting in the node for multi-grained scanning during random forests training
min_samples_cascade	5	The minimum number of samples to perform splitting in the node for cascading random forests training

4.4 Comparison experiments and results

4.4.1 Data balance

It needs to pre-process the sample set because the crawled text samples are unbalanced. The number of samples is reduced by using a down-sampling algorithm for those commodity texts with many samples. In contrast, data augmentation algorithms are used to increase the number of samples for those commodity texts with a small number of samples. It can be referred to as Sect. 3.2.

The effect of using data balancing operations on the model's accuracy is shown in Fig. 12. It can be seen in Fig. 13, after the data balancing operation, the Precision value, Recall value, and the F_1_score value of the model on this dataset are all significantly better than that on the original dataset (both the training set and validation set).

The effects of using data balancing operation on the model training process are shown in Figs. 13 and 14. It can be seen that the model converges faster when trained on the balanced data set, the value of the loss is smaller, and the curve on the training set fits better to the curve on the validation set.

4.4.2 Word embedding comparison

We use the Bert model, the Word2Vec model, and the Word2Vec model with the optimal window mechanism to extract text features, respectively.

The classification accuracies of these three models with the same classifier are shown in Table 4. It should be noted that the classifier uses only the default parameter settings. After adjusting the parameters, the classifier accuracy will be improved in the final experiments. It is well known that the representation of text features will significantly impact the classification accuracy of subsequent classifiers. As shown in Table 4, the feature representations of the commodity description text using the Bert model and the Word2Vec model have

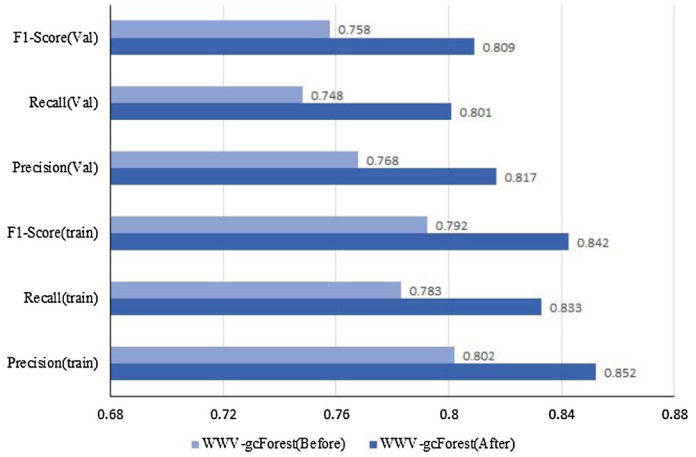

Fig. 12 The effect of data balancing operation on the accuracy of the model

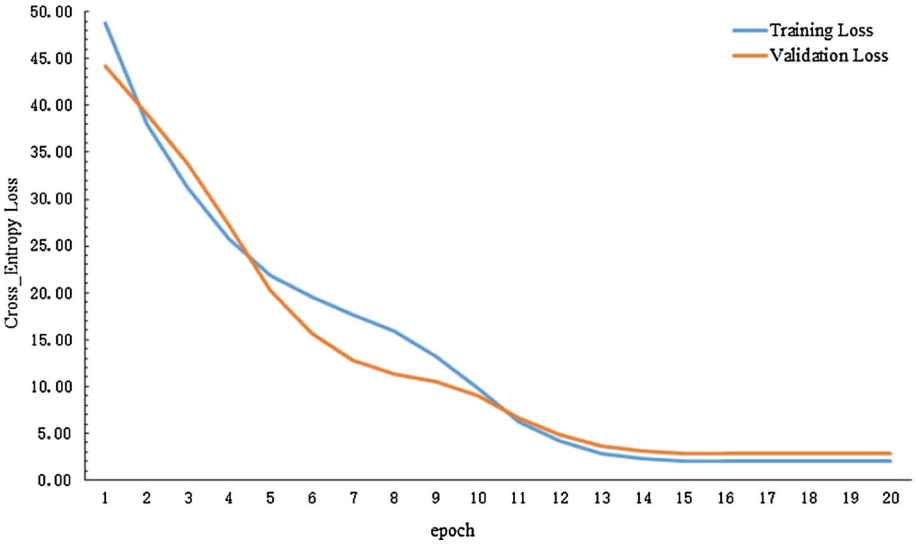

Fig. 13 Before the data balancing operation

similar effects. The fine-tuned Bert model and Word2Vec models with the optimal window mechanism also have approximate feature representations of the commodity description text. Overall, the Word2Vec model with the optimal window mechanism is better than the other models.

We randomly selected five commodity categories. Ten commodity description texts were randomly selected in each category, and these texts were embedded respectively by using these three models. The feature vectors represented by various models are decomposed by using the PCA algorithm and then visualized to display. The Figs. 15, 16, and 17 show that each point means a sentence vector, and each color indicates a commodity category.

After visualizing these feature vectors, we can find that the vectors generated by the Bert model and Word2Vec model, mapping intervals are relatively concentrated. While the vector

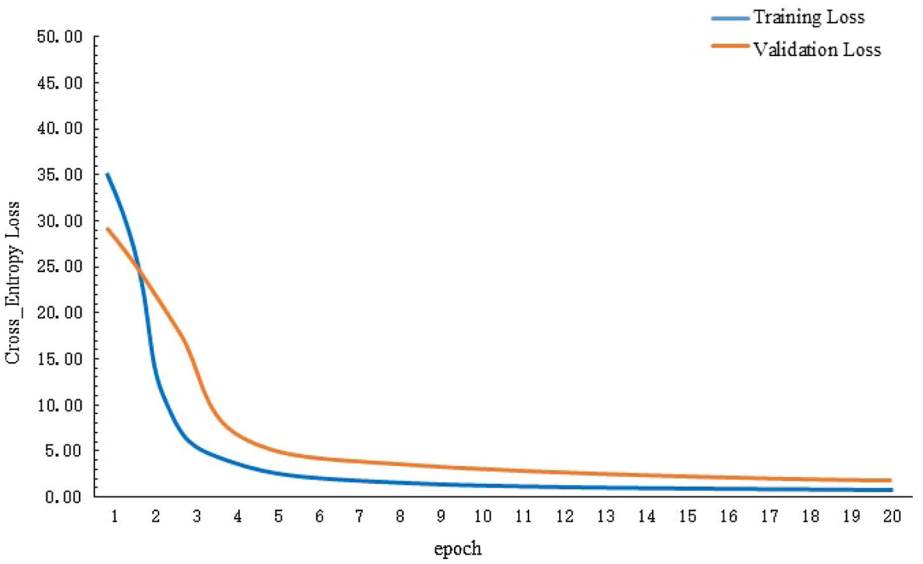

Fig. 14 After the data balancing operation

Table 4 Word embedding comparison of four models

Model name	Precision	Recall	F_1_score
Bert-gcForest	0.647	0.627	0.637
Bert (fine_tunning)-gcForest	0.711	0.703	0.707
Word2vec-gcForest	0.658	0.615	0.636
Word2vec-window-gcForest	***0.722***	***0.709***	***0.715***

The bold italics values highlight the model results used in this paper

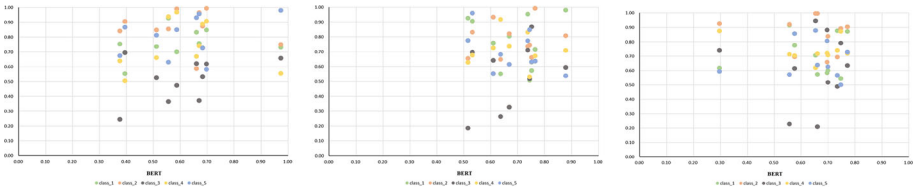

Fig. 15 Visualization of BERT-generated vectors in 3 times sampled

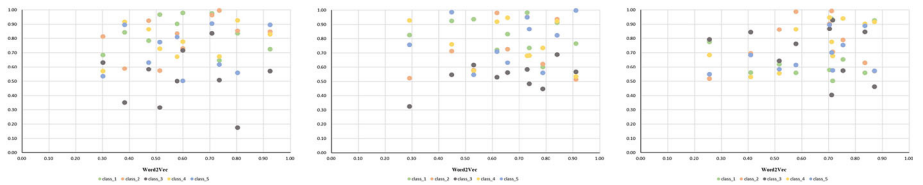

Fig. 16 Visualization of Word2Vec-generated vectors in 3 times sampled

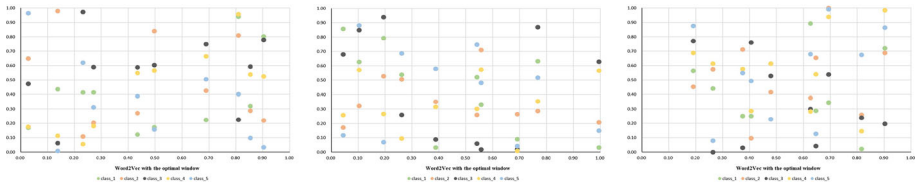

Fig. 17 Visualization of Word2Vec with the optimal window generated vectors in 3 times sampled

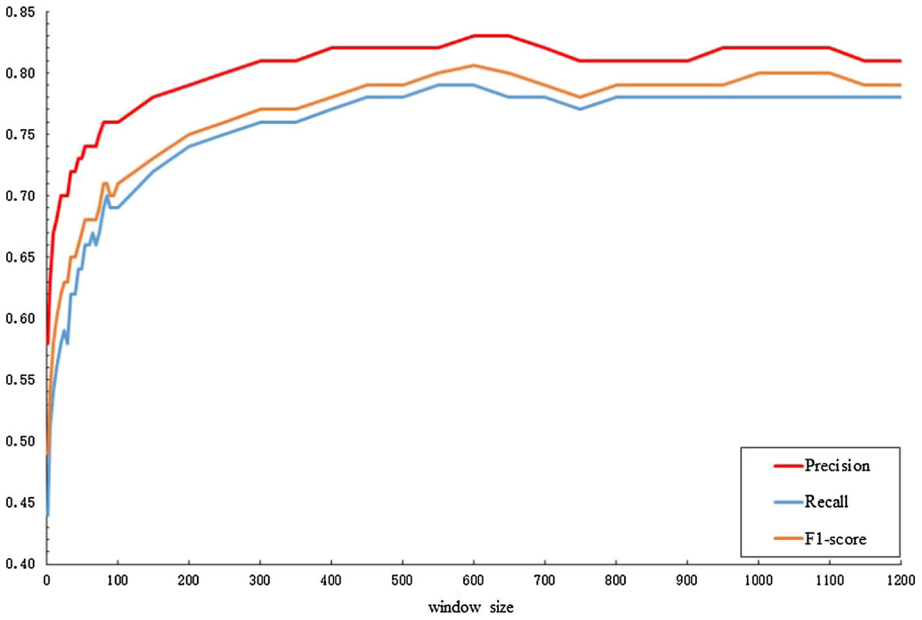

Fig. 18 The influence of the "optimal window" value on P, R, and F_1_score values

mapping intervals generated using the Word2Vec model with the optimal window mechanism are relatively scattered. The more significant the difference between the various commodity feature vectors, the less difficult it is to classify them. Then the embedding representation of the commodity description text based on the Word2Vec model with the optimal window mechanism is more beneficial to reduce the classification difficulty of subsequent classifiers.

So, we studied the effect of "window size" on the accuracy of the classification model and analyzed. The effects of the "window size" on precision, recall, and F_1_score are shown in Fig. 18. As shown in Fig. 18, similar commodity description short texts are used in training with the window size increasing. The precision and recall values have significantly improved. When the window size is 600, the precision, the recall, and the F_1_score values reach their optimal value, and then they will tend to be stable. Accordingly, the optimal window size was chosen to be 600.

The optimal window mechanism significantly improves the quality of text embedding. The reasons are as follows.

Firstly, in the CBOW module, known from Eq. 10, the current word encoding is determined by the Huffman tree and the conditional probability of the co-occurring words before and after it. Under the optimal window mechanism, the current short text content was expanded by

similar texts. The counting range (window) of co-occurrence words enlarged before and after the current word. The number of words involved in calculating the conditional probability with the present word increased as the window size increased.

The current word conditional probability is the product of the conditional probabilities of all its co-occurring words. Because the conditional probability value is not greater than 1, the value will become smaller after multiplying by all the conditional probability. As the number of co-occurring words increases, the smaller the conditional probability of the current word, the closer the corresponding node in the Huffman tree will be to the leaf node, which means that the depth of the Huffman tree will be larger. As a result, the length of the path in the Huffman tree to the leaf node where the current word is located will also increase, and the number of generated Huffman code bits will increase. Since the number of bits encoded for the current word increases, the difference between the feature vectors increases, which reduces the difficulty of the subsequent classification task.

Secondly, these commodity short-texts characterize by confusing word order, a weak association between words, and near-synonyms intensive occurrence. These characteristics make the Bert model not perform well, but the Word2Vec model based on the conditional probability calculation of word co-occurrence performs better. In particular, the application of pre-clustering and the optimal window mechanism makes the conditional probability calculation of current words more accurate.

4.4.3 Visualization of multi-headed attention

It is well known that the patches at the current location play a dominant role in Self-Attention. At the same time, the Multi-Headed Attention mechanism enables the model to focus on patches at multiple locations in the image, which improves the processing performance of the Attention layer more effectively.

To verify the role of the Multi-Headed Attention mechanism in the model, we selected some images to visualize the attention regions of the images generated by the Multi-Headed Attention mechanism, as shown in Fig. 19. The regions with high brightness are the regions with larger attention value allocation, while those with low brightness are those with smaller attention value allocation.

It is easy to see from Fig. 19 that patches containing commodities in the image are assigned to higher weight values and appear bright, while other patches are set lower weight values and appear dark.

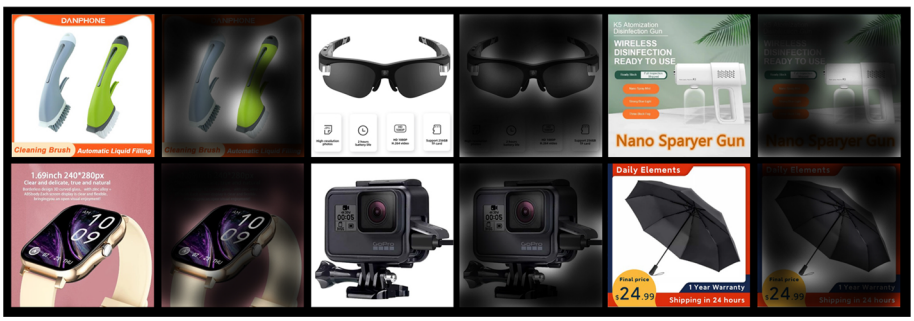

Fig. 19 The multi-headed attention mechanism visualization

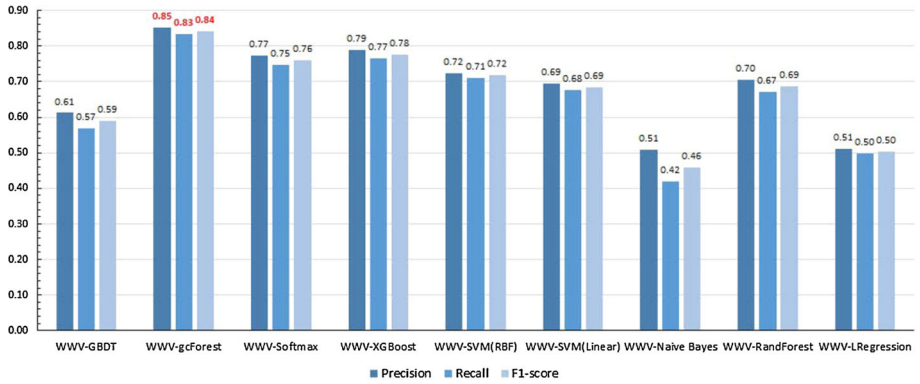

Fig. 20 The classifier comparison experiments

4.4.4 Selection of classifier

It is necessary to select a suitable classifier model for the classification tasks, and it will be helpful to improve the classification accuracy. Many models are commonly used as classifiers, such as the Softmax model, the SVM model, the Logistic Regression model, the XGBoost model, and so on. The same multi-modal vector is used as the input vector, and different classifiers are used as classification tools, respectively, and the experimental results are shown in Fig. 20. The experimental results show that the gcForest model performs best when compared with others.

4.4.5 Ablation experiments

The WWV-gcForest multi-modal model performs better than others through previous experiments. Next, we will use ablation experiments to reveal the contributions of each modal feature vector and the optimal window mechanism. The text features, the image features, and the optimal window mechanism will recombine with the same classifier, and the ablation experiment results are shown in Fig. 21.

From the experimental results, it is easy to see that the model's accuracy based only on image features is the lowest. These models with fusion features generally perform better than those single-feature models. The addition of the optimal window mechanism plays an optimized word embedding role, which helps to improve model accuracy. Also, since the optimal window mechanism increases the difference between vectors, it significantly improves the model's accuracy.

In terms of their contribution to improving the model accuracy, the semantic text features contribute the most, followed by image features. At the same time, the optimal window mechanism plays an icing on the cake.

4.5 Experimental results

The classical neural network models are selected to have some comparison experiments, and the experimental results are shown in Table 5.

As shown in Table 5, the multi-modal composite feature model with the optimal window mechanism gives the best results. The results also prove the robust performance of the ViT

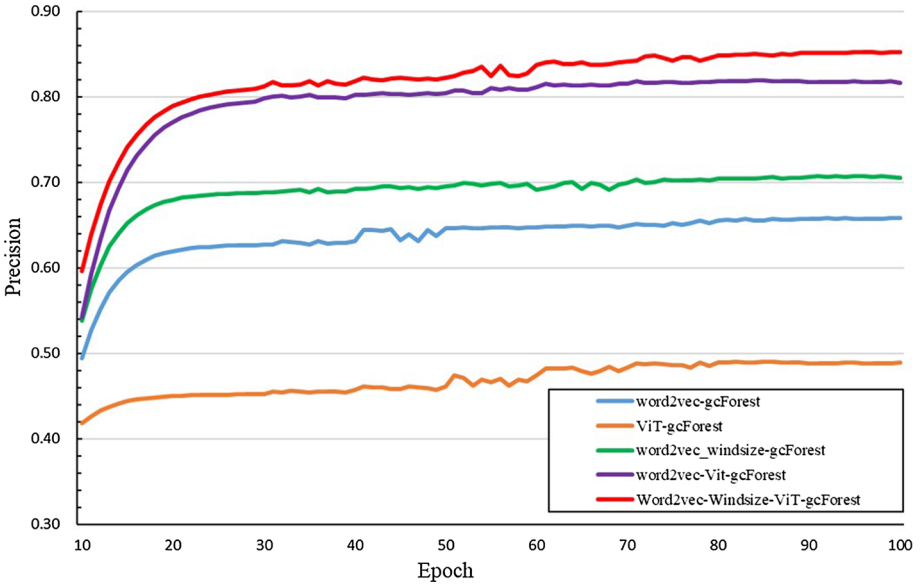

Fig. 21 The WWV-gcForest model ablation experiments

Table 5 The experiment results

Model name	Precision	Recall	F_1_score
WWAlexNet-gcForest	0.672	0.641	0.656
WWGoogleNetv4-gcForest	0.779	0.752	0.765
WWResNet50-gcForest	0.835	0.812	0.823
WWDenseNet-gcForest	0.838	0.823	0.831
WWVGG19-gcForest	0.692	0.636	0.663
WWViT-gcForest	***0.852***	***0.833***	***0.842***

The bold italics values highlight the model results used in this paper

model in extracting image features. It also shows the reasonableness and correctness of the ViT model selected to extract image features.

5 Conclusion

The research goal of this paper is to optimize the customs declaration process of cross-border e-commerce commodities and solve the issue of intelligent recognition commodity information in the customs declaration process. Selecting commodity names as the research object, we use the Word2Vec model with the optimal window mechanism and the ViT model extract short text features and commodity image features, respectively. Generate multi-modal composite features and then complete the name recognition task using the deep cascade forest model.

Our proposed multi-modal name recognition model with the optimal window mechanism can solve the commodity name recognition issue better. The precision is 0.85, and the recall is 0.83. The proposed optimal window mechanism increases the difference between feature

vectors and reduces the difficulty of the classification task, which is a novel idea of feature optimization. Also, the ViT model performs better compared to the classic CNN models when dealing with large image data sets. Finally, we also verified the correctness and effectiveness of our model through ablation experiments and comparison experiments.

In the future, we will enrich the commodity categories, increase the sample data, and optimize the hyper-parameters to improve the accuracy.

In conclusion, the multi-modal name recognition model based on the optimal window mechanism is novel, reasonable, and effective. The model has high practical application value and can provide an effective solution for intelligent customs declaration, improve the accuracy of information recognition, enhance efficiency, decrease costs and optimize the process. Meanwhile, the model can also be used for named entity identification, rumor classification, mail classification, and many other fields. In addition, this study can provide empirical and algorithmic references for scholars interested in this field.

Acknowledgements This study was sponsored by the "National Natural Science Foundation of China" (72174086).

References

Alfaro, C., Cano-Montero, J., Gómez, J., Moguerza, J. M., & Ortega, F. (2016). A multi-stage method for content classification and opinion mining on weblog comments. *Annals of Operations Research, 236*(1), 197–213.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Burdick, L., Kummerfeld, J. K., & Mihalcea, R. (2021). Analyzing the surprising variability in word embedding stability across languages. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 5891–5901).

Chen, L., Chou, H., Xia, Y., & Miyake, H. (2021). Multimodal item categorization fully based on transformer. In: *Proceedings of the 4th workshop on e-commerce and NLP* (pp. 111–115).

Chen, Y. (2015). *Convolutional neural network for sentence classification*. Ph.D. thesis, University of Waterloo.

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on empirical methods in natural language processing (EMNLP 2014)*

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297.

CSY. (2020). *China statistical yearbook*. China Statistical Publishing House.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S., et al. (2020). An image is worth 16 × 16 words: Transformers for image recognition at scale. In *International conference on learning representations*.

Guo, B., Zhang, C., Liu, J., & Ma, X. (2019). Improving text classification with weighted word embeddings via a multi-channel TextCNN model. *Neurocomputing, 363,* 366–374.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Joulin, A., Grave, E., & Mikolov, P. B. T. (2017). Bag of tricks for efficient text classification. *EACL, 2017,* 427.

Kenton, J. D. M.-W. C., & Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT* (pp. 4171–4186).

Kumar, A., Singh, J. P., Dwivedi, Y. K. et al. (2020). A deep multi-modal neural network for informative Twitter content classification during emergencies. *Annals of Operations Research*. https://doi.org/10.1007/s10479-020-03514-x.

Kumar, H., Agarwal, A., Dasgupta, R., & Joshi, S. (2018). Dialogue act sequence labeling using hierarchical encoder with CRF. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32).

Luo, Y., Ma, J., & Li, C. (2020). Entity name recognition of cross-border e-commerce commodity titles based on TWS-LSTM. *Electronic Commerce Research, 20*(2), 405–426.

Ma, J., Li, X., Li, C., He, B., & Guo, X. (2019). Machine learning based cross-border e-commerce commodity customs product name recognition algorithm. In: *Pacific Rim international conference on artificial intelligence* (pp. 247–256). Springer.

Mikolov, T., Yih, W.-T., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 746–751).

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543).

Sun, F., & Chen, H. (2018). Feature extension for Chinese short text classification based on LDA and word2vec. In *2018 13th IEEE conference on industrial electronics and applications (ICIEA)* (pp. 1189–1194). IEEE.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., & Tang, X. (2017). Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156–3164).

Zhou, Z.-H., & Feng, J. (2019). Deep forest. *National Science Review, 6*(1), 74–86.

Zhu, T., Wang, Y., Li, H., Wu, Y., He, X., & Zhou, B. (2020). Multimodal joint attribute prediction and value extraction for e-commerce product. arXiv preprint arXiv:2009.07162