IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 21, 2020, accepted March 30, 2020, date of publication April 2, 2020, date of current version April 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985098

Semi-Supervised Multi-Granularity CNNs for Text Classification: An Application in Human-Car Interaction

FEN ZHAO[1], **YINGUO LI**[1,2], **LING BAI**[1,3], **ZHEN TIAN**[1,3], **AND XINHENG WANG**[4]

[1]School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2]School of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[3]Department of Electrical and Computer Engineering, McMaster University, Hamilton L8S 4L8, Canada
[4]School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Corresponding author: Xinheng Wang (xhwangcqupt@gmail.com)

This work was supported in part by the Doctoral Candidate Training Program of Chongqing University of Posts and Telecommunications under Grant BYJS201916.

ABSTRACT With the rapid growth of advanced driving assistance technology, how to implement human-car interaction humanely and personally becomes increasingly important. Existing methods typically rely on hand-crafted data to conduct training, which cannot fully extract information hidden in text. In this paper, we tackle the problem of label data acquisition difficulties and inadequate feature extraction via semi-supervised learning and multi-granularity learning. To fulfill the goal of label data acquisition and feature extraction, we present a novel semi-supervised multi-granularity convolutional neural networks (CNNs)-based (SSMGCNNs-based) model for an application in human-car interaction, which consists of the two-view-embedding (TVE) module and the multi-granularity CNNs (MGCNNs) module. The TVE module learns embeddings of text regions from the unlabeled user command data set and then integrate the learned tv-embeddings into MGCNNs, so that the learned tv-embedded regions are used as an additional input into MGCNNs' convolution layer to solve the problem of data annotation. MGCNNs can fully extract information hidden in text by multiple convolution kernels of the same convolution layer. We compared our model with some state-of-the-art machine learning models. On the car operation command data set, the simulation results demonstrated that, compared with CNNs, our method respectively improved 5.13%, 5.64%, 3.60% and 5.34% on precision, recall, F-1 and training loss.

INDEX TERMS Convolutional neural networks (CNNs), human-car interaction, two-view-embedding, multi-granularity.

I. INTRODUCTION

With the development of affordable sensing and computing technologies, various intelligent technologies have been commercialized recently [1], [2]. One of the most popular fields of intelligent technologies is the assistance driving field [3], which focuses on improving the autonomy in the transportation system. In order to achieve human-level perception reasoning and driving policies in the field of assistance driving, a great deal of work has been done [4].

At present, in-car intelligent technologies are becoming ubiquitous. Since many valuable services are provided for human drivers, which meets service demands of intelligent interaction, human drivers expect to use these in-car intelligent applications to implement human-car interaction [5], [6]. In addition, due to traffic jams, people spend significant time in their cars. Cars are more than means of transport, and people value their cars as personal spaces, in which people can entertain themselves by the intelligent interaction system [7]. Besides in-car entertainment functions, cars provide auto control functions [8], [9]. It is common that the driver operates a car, meanwhile, he can interact with in-car devices. To be more specific, when the driver operates a car, he can wake up in-car intelligent voice systems to call a friend, look up an address from the navigation system or interact with other services, as shown in FIGURE 1. Thus, as an important part of the research in the assisted driving field, human-car interaction technology plays a crucial role

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

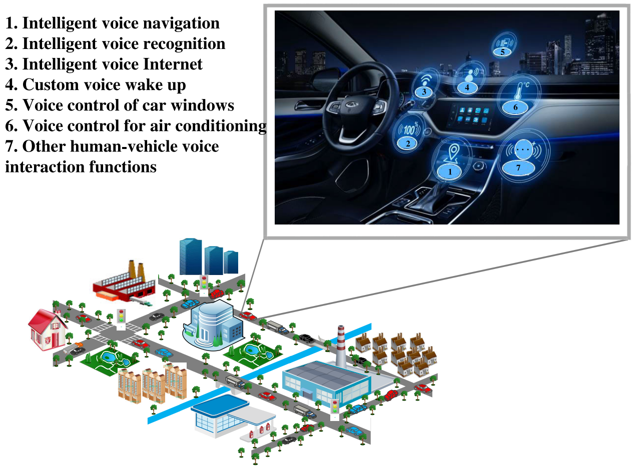

1. Intelligent voice navigation
2. Intelligent voice recognition
3. Intelligent voice Internet
4. Custom voice wake up
5. Voice control of car windows
6. Voice control for air conditioning
7. Other human-vehicle voice interaction functions

FIGURE 1. The typical in-car intelligent voice service.

in the guidance of the intelligent transportation system, which makes the research of improving interaction operations particularly urgent.

Existing methods of assisted driving are almost supervised algorithms [10], which use labeled data to train models. It is well known that labeled data has characteristics of acquisition difficulties. The limitations of labeled data pose a huge challenge to assisted driving. In order to address the above-mentioned impediment, many contributions have been made by researchers. For large-scale learning tasks, lots of unlabeled data are unused unless using some manually designed rules. These particular manual efforts increase labor costs and time expenditure. However, semi-supervised learning can alleviate the problem of data annotation. In the field of semi-supervised learning, Zhou proposed an ensemble learning-based Co-Forest algorithm [11]. However, the Co-Forest algorithm simply selects unlabeled data of the high confidence to be added to labeled sets, while all low confidence unlabeled data is wasted. Active learning has improved the disadvantage [12]. Considering that low confidence unlabeled data may have more helpful information as well, labels of low confidence data can be provided by domain experts such as manual annotation [13], [14].

On the other hand, based on inadequate extraction of feature information, many methods that extract feature information are established. However, some existing shallow methods typically suffer from over-fitting, which makes a promising solution based on deep neural networks (DNNs) widely concerned in feature extraction. One of the most popular methods based on DNNs is convolutional neural networks (CNNs), which have become research hot spots in the fields of computer vision, natural language processing and voice recognition [15], [16]. In the application of CNNs, the original input such as images can be directly inputted without complicated data preprocessing. Based on this idea, it is of great research value to process text data of the driver operation command into a matrix similar to images and use CNNs to build the classification model.

In our work, we focus on the classification of the driver operation command, in which the amount of labeled data used for classification is limited and feature information is extracted poorly. In this paper, we present a novel semi-supervised multi-granularity CNNs-based (SSMGCNNs-based) method, which consists of the two-view-embedding (TVE) module and the multi-granularity CNNs (MGCNNs) module. The method adopts the TVE module to learn embeddings of text regions from the unlabeled user command data set, which alleviates the problem of label data acquisition difficulties. In addition, MGCNNs are used to fully extract information hidden in text data to improve performance of human-car interaction.

We explore the typical intelligent application scenario of automobiles to complete the functions such as drivers intention reasoning, information interaction and the autonomous control of automobiles. Intelligent navigation, intelligent entertainment, intelligent information interaction and other functions of automobiles are realized. In addition, a new automobile design point from the perspective of artificial intelligence is put forward, which greatly improves the automobile functions in the current market and enables drivers to enjoy the high-quality driving experience.

In a nutshell, we make the following contributions.

- To alleviate the problem of label data acquisition difficulties, the semi-supervised learning model is proposed, which trains the TVE model to obtain labeled data from unlabeled data.
- To improve the feature extraction capability, we modify the internal structure of CNNs and propose MGCNNs to fully extract information hidden in text by multiple convolution kernels of different sizes in the same convolution layer.
- Compared with existing researches, our proposed model gives improved performance, which is proved by extensive experiments. In terms of precision, recall, F-1 and training loss, our model improved 5.13%, 5.64%, 3.60% and 5.34% compared with CNNs, respectively.

The rest of the paper is organized as follows. Section II reviews related works. The problem formulation and challenges are described in Section III. In Section IV, we present the human-car interaction algorithm based on SSMGCNNs. The performance of SSMGCNNs algorithm is evaluated through extensive experiments in Section V, followed by the conclusion in Section VI.

II. RELATED WORK

Car operation command data has characteristics of limitations, which make the classification of the car operation command with obvious instability and uncertainty. Thus, taking effective measures to tackle the problem of label data acquisition difficulties and inadequate feature extraction is particularly important yet challenging in the classification of the car operation command. So far, many researches of text classification, each with its own advantages and limitations, were proposed. The most commonly used classification models include linear support vector machine (SVM) model, Bayesian network model, back propagation network (BPN) model, neural network (NN) model and some DNNs models. Owing to the effectiveness of label samples acquisition and the sufficiency of feature extraction, the DNNs-based methods have become research hot spots in these classification methods.

Recent years have witnessed the significant development of DNNs models, which have played a significant role in natural language processing (NLP). Since intrinsic features and feature extraction are integrated by the DNNs-based model, it has been proven useful for many NLP tasks such as text classification. Xu and Cai [17] constructed a model to incorporate context-relevant information into CNNs for text classification. In their work, two hidden layers were used to extract concept and context features respectively, and then an attention layer was employed to extract those context-relevant concepts. Next, CNNs was utilized to extract high-level features from words and these context-relevant concept features. Yao *et al.* [18] proposed a new knowledge-guided DNNs model for effective disease classification. Firstly, trigger phrases were recognized; then, classes were predicted by very few examples; next, trigger phrases were used; finally, CNNs were trained by word embeddings of medical language system. He *et al.* [19] explored evolutionary neural networks for time-evolving text classification. A simple way was introduced to extend arbitrary neural networks to evolutionary learning by using a temporal smoothness framework, and then a diachronic propagation framework was proposed to incorporate the historical impact into currently learned features through diachronic connections. Zheng and Zheng [20] proposed a bidirectional recurrent convolutional neural network-based model to address the problem of the multi-class text classification. To implement the goal of fine text classification, the bidirectional long short-term memory network and CNNs were combined with the attention-based model. In this work, a bidirectional recurrent network was applied to extract relevant information. Important words can be inferred by the maximum pooling layer of CNNs, and higher weights can be given by the attention-based model. Banerjee *et al.* [21] proposed CNNs Word Glove and Domain phrase attention-based hierarchical recurrent neural network (DPA-HNN) for synthesizing information on pulmonary emboli (PE) from over 7370 clinical thoracic computed tomography (CT) free-text radiology reports. In this work, domain-dependent phrases were encoded into an attention mechanism, and then a radiology report was represented through a hierarchical RNN structure.

III. PROBLEM FORMULATION AND CHALLENGES

In this section, the problems of existing solutions are described. Considering these problems, a promising solution is proposed in the design of our study.

A. PROBLEM DESCRIPTION

Recent years have witnessed the significant application requirement of NLP [22]. Humanized services will be provided by practical NLP technology. The DNNs-based model has been proven useful in many NLP tasks such as text classification owing to the integration of intrinsic features and feature extraction. However, existing approaches of text classification are almost supervised algorithms [10], which use labeled data to train models [23]. It is well known that labeled data is difficult to obtain. In addition, based on the complexity of data, a number of methods are built to extract feature information. However, most of existing shallow systems typically suffer from over-fitting, which cannot fully extract information hidden in text.

B. WORKING MECHANISM OF THE INTELLIGENT SYSTEM

Automatic text classification has become a key technical task in the human-car interaction system. Moreover, the DNNs model can actively learn grammatical features and semantic features, which is feasible for text classification tasks [24]. However, existing supervised DNNs models use labeled data to train the classifier model, and labeled data needs to be annotated by human experts which costs a lot of manpower and resources. Aiming at the problem of data annotation, this paper studies the application of semi-supervised learning mechanisms in CNNs. A two-view-embedding (TVE) model is proposed to learn labeled data from unlabeled data, which reduces labor cost in data annotation. In addition, in order to solve the phenomenon of too many parameters caused by too many hidden layers in the CNNs model, the internal structure of the network with local perception and weight sharing is proposed. However, when only a fixed convolution kernel is used for operation, it doesn't fully extract feature information hidden in data. Therefore, the MGCNNs-based model is proposed, which extracts more feature information hidden in text. The implementation details of the proposed model include three aspects. Firstly, the one-hot-vector (OHV) model is used to learn vector representations of user commands. Secondly, the TVE model learns embeddings of text regions from the unlabeled user command data set. Thirdly, embeddings of text regions are integrated into MGCNNs, so that embeddings of text regions can be used as an additional input to the convolution layer of MGCNNs.

C. CHALLENGES

In order to provide drivers with a more humane driving experience from in-car functions to driving control, some problems should be considered in the implementation of the DNNs-based interaction system, i.e., how to alleviate the problem of label data acquisition difficulties in the human-car interaction system, and how to fully extract information hidden in text to accurately execute the car function. In this paper, we focus on how to address the two challenges.

- For the text classification task based on supervised algorithms, a large number of label data needs to be provided.

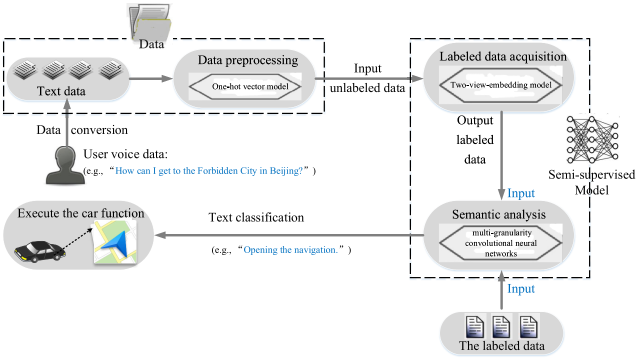


FIGURE 2. The overview of the proposed SSMGCNNs model.

However, label data can only be obtained by manual annotation.

- We note that the CNNs-based models have realized better intelligent functions than other existing models, but a huge challenge remains on the internal structure of the network - such as only using a fixed convolution kernel to extract feature information in the convolution operation.

Note that, more expressive vector representations of user command data are the prerequisite for executing the car function and interacting with the car. Before we implement the human-car interaction, we first implement the valid vector representation to better predict car functions that the user may need in the current environment.

IV. METHODS

The overview of our framework is shown in FIGURE 2. Firstly, the model converts the speech command into text. Secondly, we use the OHV model to learn vector representations of the text command. The OHV model directly learns vector representations of small text regions. Thirdly, the TVE model is trained to assign a label for each small text region of size m, and the learned label data is used as part of the MGCNNs' input. Next, the function that the user may need in the current context is predicted, and then the system executes the car function. Finally, the system interacts with the human driver. The specific details for SSMGCNNs-based text classification task can be given in Algorithm 1. (Note that take "open the music" as an example.)

A. PRELIMINARY: ONE-HOT-VECTOR MODEL FOR DATA PREPROCESSING

In fact, the data is not always a continuous value. In most cases, it is a discrete value such as gender or category. Based on these observations, a promising solution to effective mathematical representation for the discrete values is introduced. We call this as the OHV model in this paper.

At present, the OHV model has been introduced into vector representation of text data. Suppose that a sentence $S = (d_1, d_2, \ldots, d_n)$ with V vocabularies is given. The CNNs-based model requires the word embedding of the sentence as the input of the model. The straightforward way is to treat the sentence S as a picture of $|S| * 1$ dimension with V channels, and each word d_i is treated as a pixel. In this sense, each word is represented as a $|V| - dimensional$ one-hot vector. Suppose that the vocabulary $V = ``close'', ``don't'', ``music'', ``open'', ``the''$ and the user command $S =$ "open the music". Then, the one-hot vector representation of the user command: $v = [00010\,|00001\,|00100\,]^T$ is given.

Algorithm 1 SSMGCNNs Training Algorithm

Require: the driver command data set

Ensure: training loss rate

```
set a threshold $\theta$;
while precision $P \le \theta$ do
    input the driver command data set;
    use one-hot-vector model to preprocess text data,
      and then obtain the bow-one-hot vector of
      the driver command
          $X = \begin{bmatrix} 00011 \\ 00101 \end{bmatrix}$
    use two-view-embedding model to predict
      adjacent area:
          $P(X_2\,|X_1) = g_1(f_1(X_1), X_2)$
    integrate the learned tv-embeddings into
      multi-granularity CNNs using Eq. (3);
    for all convolution layers in MGCNNs do
        the filter performs the convolution operation;
        get different degrees of feature dictionaries $D$;
    end for
    for all pooling layers in MGCNNs do
        perform maximum pooling operation;
        generate a set of univariate feature vectors;
    end for
    for softmax layer in MGCNNs do
        classify the car operation command using
          Eq. (6);
    end for
    return training loss rate;
end while
input validation set $V$, and then adjust
    classifier parameters, as shown in FIGURE 15;
input test set $E$, and then test the classification
    capacity of the model, as shown in FIGURE 16.
```

Now, we briefly discuss the application of the seq-one-hot vector model and the bow-one-hot vector model in text classification [25], [26]. We train each small text region to obtain region embedding. For instance, on the user command S above, we would have two small text regions $``open\quad the''$ and $``the\quad music''$ with $r = 2$ and stride $= 1$. The two small text regions are represented by the following vectors: $r_0(v) = \begin{bmatrix} 00010 \\ 00001 \end{bmatrix}$ $, r_1(v) = \begin{bmatrix} 00001 \\ 00100 \end{bmatrix}$, respectively. We

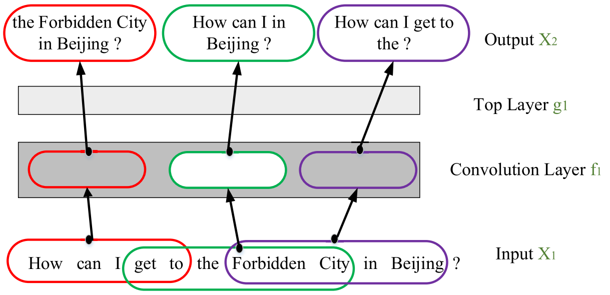


FIGURE 3. Two-view-embedding (TVE) learning to predict the context (region size = 5, stride = 3).

call it seq-one-hot vector ('seq' for keeping the sequence of words). Considering the vector dimensionality, we note that the dimensionality of each region vector $r_i(v)$ could be very high if the number of the vocabulary V is very large (e.g., 100K) in the seq-one-hot vector way. Moreover, if the dimensionality of the region vectors is too high, the training model could become very complex, so that, more parameters would be learned. Therefore, the low-dimensional region vector is desirable. The bow-one-hot model is a promising way to solve the problem of high dimensional data representation. As an example, the region vector of the user command (namely, "open the music") is converted to: $r_0(v) = [00011]$, $r_1(v) = [00101]$, where the $r|V| - dimensional$ region vector is converted to $|V| - dimensional$ vector. Based on these observations, the bow-one-hot vector representation is used to alleviate the problem of high-dimensional data representation in the paper.

B. SEMI-SUPERVISED MODEL FOR TEXT CLASSIFICATION

The semi-supervised model includes the following two steps: TVE learning and supervised learning. The semi-supervised model directly learns embeddings of small text regions from unlabeled data with the TVE model, as illustrated in FIGURE 3. And then, the learned embedding is integrated into supervised CNNs.

The TVE model computes the embedding representation of the small text region via:

$$P(X_2|X_1) = g_1(f_1(X_1), X_2), \ (X_1, X_2) \in \chi_1 * \chi_2 \quad (1)$$

where X_1 is the input vector, X_2 is the predictive value, f_1 is represented by the convolution layer which is a tv-embedding of $\chi_1 \quad w.r.t \quad \chi_2$, and g_1 is embodied by the top layer. Based on the definition, we note that X_1 retains all the content required for prediction X_2.

we aim to address this issue of label data acquisition difficulties. Based on this fact, a task on unlabeled data is created to assign a label for each small text region. Since CNNs build up from these small text regions to implement text classification, a sub-task that assigns a label for each small text region is considered instead of the ultimate task of classifying the entire sentence. In this sense, the TVE learning model is sensible on the semantic analysis task. In the user command "How can I get to the Forbidden City in Beijing?", the label of "How can I get to" is obtained from "How can I get to" itself ($view - 1 : X_1$) and its adjacent text region "the Forbidden City in Beijing?" ($view - 2 : X_2$). We train the TVE model to predict X_2 from X_1. $P(X_2|X_1)$ is approximate by $g_1(f_1(X_1), X_2)$, as shown in Formula 1.

The TVE model consists of an input layer, a convolution layer, a top layer and an output layer. The convolution layer is composed of neurons, each of which connects to a small part of a sentence. These small parts collectively cover the entire sentence. Given a sentence S, for the $\kappa - th$ text region, we compute the convolution layer of TVE model as follows:

$$T_\kappa(s) = \sigma^{(T)}\left(W^{(T)} * r_\kappa^{(T)}(s) + b^{(T)}\right) \quad (2)$$

where σ is the activation function. In the paper, we use rectified linear unit (ReLU) activation function. $r_\kappa^T(s) \in \Theta^q$ is the vector representation of the $\kappa - th$ text region. $T_\kappa(s)$ is used as the input of the top layer. Parameters W^T and b^T are shared by neurons in the same layer.

Next, the tv-embedding learned from unlabeled data by the TVE model is integrated into supervised CNNs. That is the tv-embedding is used as part of CNNs' convolution layer input. Hence, we compute the computation unit of CNNs' convolution layer by replacing $\sigma(W * r_\kappa(s) + b)$ with:

$$\sigma(W * r_\kappa(s) + V * T_\kappa(s) + b) \quad (3)$$

The weight matrix W and V, bias vector b and the top-layer parameters are updated, so that the designated loss function is minimized on the labeled data.

C. THE CONSTRUCTION OF MULTI-GRANULARITY CONVOLUTIONAL NEURAL NETWORKS

When too many hidden layers are constructed in the CNNs, the phenomenon of dimensional disaster will occur. Thus, the internal structure of local perception and weight sharing is proposed [27]. However, when only a fixed convolution kernel (namely, region size) is used during operation, the feature information hidden in the text data cannot be extracted sufficiently. Aiming at the problem of inadequate extraction of feature information, MGCNNs are proposed to better express text semantic information and improve classification accuracy. Through the adjustment of internal structure, feature extraction is carried out simultaneously on the same convolutional layer. Then, feature information is spliced and recombined to obtain feature information of different dimensions simultaneously.

Three kinds of region sizes (namely, S1, S2 and S3) are used simultaneously on the same convolutional layer, as shown in FIGURE 4. Each region size extracts a kind of feature information, that is, the outputs are $h_1, \quad h_2 \quad and \quad h_3$. Then, the output of the convolution layer is connected to $h = (h_1 h_2 h_3)$ to represent the entire sentence.

Each text region in the sentence is represented as a embedding region. The entire sentence with m vocabularies is

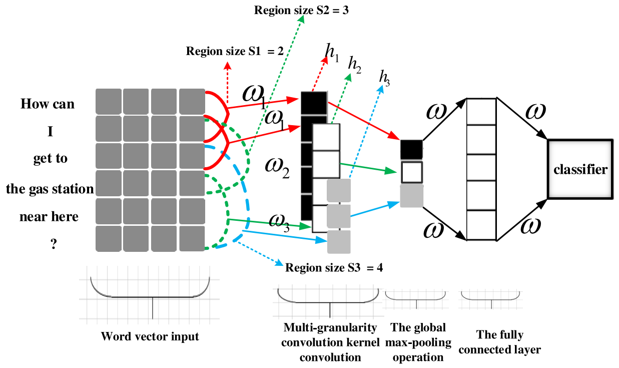


FIGURE 4. The structure chart of multi-granularity convolutional neural networks. Take "Open the navigation." as an example.

represented as:

$$S = \varsigma_1, \varsigma_2, \ldots, \varsigma_n \tag{4}$$

where ζ_i is the m-dimensional embedding of the $i-th$ text region. After convolution operation, multiple feature maps are generated, each of which is composed of a number of neurons. The neuron γ_i is expressed as:

$$\gamma_i = t\left(pooling\left(\sigma\left(\omega_i * S + b\right)\right)\right) \tag{5}$$

where t and σ represent the activation functions tanh and sigmoid, respectively. ω_i is the weight parameter, and b is the bias parameter. *pooling* represents the pooling operation. The maximum pooling is applied to select the most representative feature information from the feature map. Next, the fully connected layer inputs these features. Finally, the SoftMax layer inputs the output of the fully connected layer. The classification probability formula of the SoftMax layer is given as follows:

$$P = \sigma\left(W_s \gamma_s + b_s\right) \tag{6}$$

where σ is the activation function sigmoid, γ_s is the output of the pooling layer, and W_s and b_s are the weight parameter and the bias parameter of the SoftMax layer, respectively.

D. MODEL TRAINING

The evaluation of human-car interaction performance uses loss function. The loss function $J(\omega, b)$ is defined as follows:

$$J(\omega, b) = -\sum \log\left(P\left(y\,|D, \omega, b\right) + \lambda\,|b|\right) \tag{7}$$

where D is the input, y is the output of the model, ω and b are the weight parameter and the bias parameter respectively, and λ is the regularization parameter of the loss function.

We use the back-propagation (BP) algorithm [28] to train the model. The errors are backpropagated from top to other layers. Derivatives are calculated to optimize ω and b. Stochastic Gradient Descent (SGD) algorithm [29] is then employed to solve the convex optimization problem. Moreover, max-norm regularization (MNR) algorithm [30] is used for the column vectors of parameter matrices.

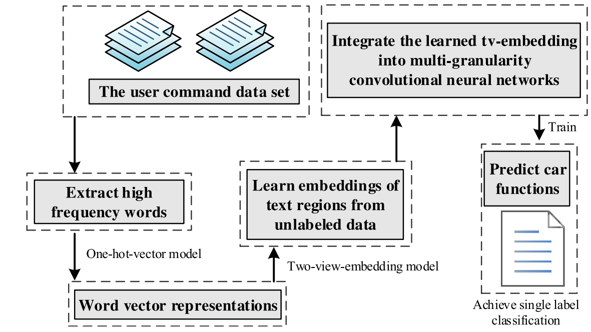

The user command data set
Integrate the learned tv-embedding into multi-granularity convolutional neural networks
Train
Extract high frequency words
Learn embeddings of text regions from unlabeled data
Predict car functions
One-hot-vector model
Two-view-embedding model
Word vector representations
Achieve single label classification

FIGURE 5. The frame diagram of human-car interaction application.

V. EVALUATION

In order to evaluate the performance of the model, we used the car operation instruction data set. This data set is a typical single label corpus. With the rapid development of artificial intelligence technology, it is very intuitive to realize human-car interaction based on NLP. When you drive a car, you can communicate with the car. To be more specific, when you want to refuel your car, you can ask how to get to a nearby gas station, then the navigation will be opened. Here, the human driver does not need to say a specific command class (e.g., opening the navigation), and he only needs to communicate with the car according to his own needs. Then, the system will carry out semantic understanding of what the driver said and predict the corresponding car function. Finally, the model executes the car function. Note that our system differs from previous work in that our system implements semantic understanding. FIGURE 5 shows the frame diagram of the CNNs-based human-car interaction.

A. DATA SETS

We used two data sets: IMDB used in [25] and the real-world car operation command data set, as summarized in TABLE 1. The car operation command data set was collected by relevant scholars. For comparative experiments, the standard data set IMDB was introduced. IMDB is a corpus of movie reviews, but it can be used to verify the model performance as well. IMDB is the sentiment classification (positive or negative) data set, which belongs to single-label classification.

Each car operation command is associated with one car function label, which belongs to single-label classification. On the single-label classification of 26 kinds of car functions, the entire corpus consist of 16000 training sets, 7000 test sets and 10000 validation sets, as shown in TABLE 1. Aiming to use as unlabeled data, 8000 items car operation command text is chosen from the same data source, so that it is disjoint from training sets, test sets and validation sets. Car functions are divided into many categories, and we mainly deal with 26 categories including opening/closing the music (represented as 1/2), opening/closing the radio (represented as 3/4), opening/closing the air conditioning (represented as 5/6), opening/closing the navigation (represented as 7/8), opening/closing the windshield wipers (represented as 9/10), opening/closing headlamps

TABLE 1. **Data sets.**

	training set	test set	validation set	unlabeled data	class	output
Car text (Chinese)	16000	7000	10000	8000	26 (single)	Car functions
IMDB (English)	25000	25000	0	75000	2 (single)	Positive/Negative (sentiment)

TABLE 2. **Experimental environment and configuration.**

Experimental environment	Environment configuration
The operating system	Ubuntu 14.04
CPU	i7-6700k
GPU	GTX 980 Ti
Internal storage	64 GB
Programming language	Python
Deep learning framework	Theano

(represented as 11/12), opening/closing fog lights (represented as 13/14), opening/closing the near-lights (represented as 15/16), opening/closing the high beams (represented as 17/18), opening/closing the left front window (represented as 19/20), opening/closing the right front window (represented as 21/22), opening/closing sunroof (represented as 23/24) and opening/closing the trunk (represented as 25/26). We set 26 kinds of car operation command as labels.

For the single-label classification on IMDB, it is a benchmark data set which is used to sentiment classification. Here, the goal is to determine if movie reviews are positive or negative. Both the training set and the test set consist of 25000 reviews, and 75000 reviews are used as unlabeled data, as shown in TABLE 1.

B. EXPERIMENTAL SETUP

In the 64-bit Ubuntu 14.04 system, we instal the experimental platform, and in the Python environment, we do model training. Configure the DNNs toolkit, firstly, the environment and Gcc are updated. Secondly, the Python running environment is built. Thirdly, the math tool: Numpy, Scipy and Theano, is installed. Then, a Theano-based DNNs package is installed. Next, scikit-neural network is installed. Again, Python test tool is installed. Finally, the display environment is initialized. The experimental configuration is shown in TABLE 2.

FIGURE 6 illustrates the human-car voice interaction and environment perception system. In this system, the understanding of driver operation commands is realized, and then the car functions are executed. In this paper, we aim to address the problem of human-car voice interaction, and the system can complete functions such as human drivers intention reasoning, information interaction and the autonomous control of automobiles.

C. CONVERGENCE ANALYSIS

The proposed SSMGCNNs method, long short-term memory (LSTM) method [31], semi-supervised method [16] and CNNs method [25] are DNNs-based methods. Aiming to obtain better classification results, the training process of the model was optimized and adjusted. In the experiment, we recorded the changes in the classification performance of the above four models at different epochs. An epoch means that one learning process of all training data is completed. The curve of training loss rate (%) with respect to the number of epochs is provided in FIGURE 7 and FIGURE 8. FIGURE 7 denotes the training loss result of different epoches on car operation command data set. As shown in FIGURE 7, we note that as the number of epochs increases, the curve becomes flat, which embodies the convergence of the DNNs-based models. Moreover, the proposed SSMGCNNs method converges faster. Analogously, FIGURE 8 denotes the training

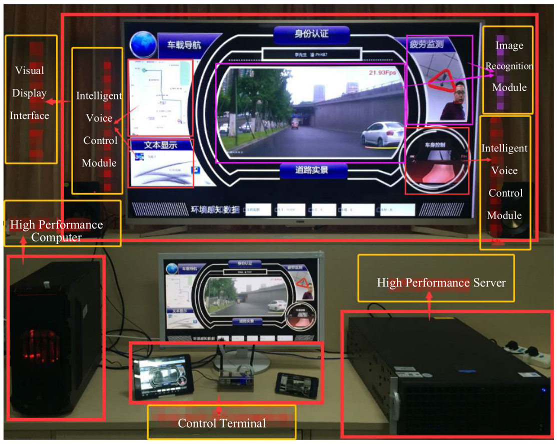


FIGURE 6. **Human-car voice interaction and environment perception system.**

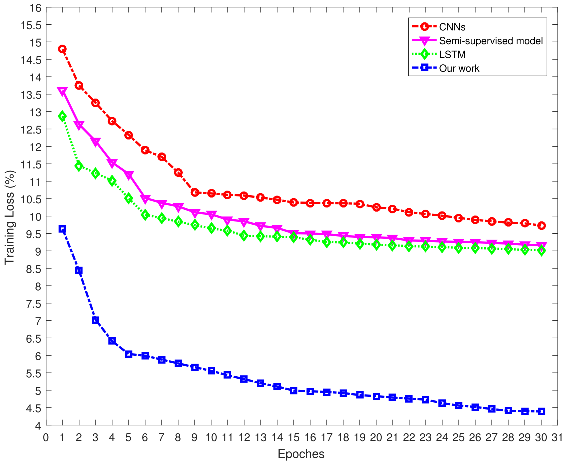


FIGURE 7. **The training loss results of different epoches in car operation command data set.**

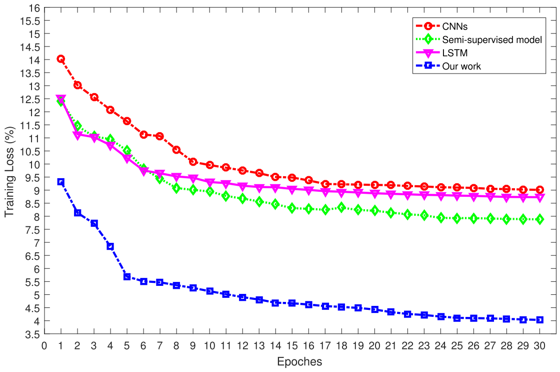


FIGURE 8. The training loss results of different epoches in IMDB data set.

loss result of different epoches on IMDB data set. Chinese characters are more difficult to process than English because there is no space between Chinese characters. The experimental results demonstrate that the training loss of the car operation command data set is slightly higher than IMDB on all models, as shown in FIGURE 7 and FIGURE 8.

D. PERFORMANCE RESULTS OF DIFFERENT MODELS

TABLE 3 shows the experimental results on the single-label classification task. As shown in TABLE 3, the first row is the linear SVM [32]; the second row is long short term memory (LSTM) [31]; the third row is CNNs [25]; the fourth row is the semi-supervised method [16]; and the last row is SSMGCNNs.

To look into the details, we experimented on different types of data sets. The first thing to note is that SSMGCNNs outperformed all baseline methods on all data sets. TABLE 3 summarizes the experimental results of all methods. Chinese characters are more difficult to process than English because there is no space between Chinese characters. Based on these observations, the experimental results demonstrated that precision of the English data set (namely, IMDB) was slightly higher than that of the car operation command data set on all models, as shown in TABLE 3. Analogously, the experimental results demonstrate that the training loss of the car operation command data set is slightly higher than IMDB.

To be more specific, SSMGCNNs (four kinds of region sizes 2, 3, 4 and 5; 700 neurons each; 9 hidden layers; followed by one unit of max-pooling each) outperforms CNNs (region size 3; 700 neurons each; 9 hidden layers; followed by one unit of max-pooling each). The results suggest that what can be learned by four kinds of region sizes is distinct enough, which complements each other. The performance of semi-supervised CNNs (region size 3; 700 neurons each; 9 hidden layers; followed by one unit of max-pooling each) also exceed the performance of CNNs.

On the car operation command data set, the best precision 95.89%, the best recall 92.75%, the best F-1 93.97% and the best training loss 4.39% were obtained by SSMGCNNs. On the car operation command data set, the experimental results demonstrated that in terms of precision, recall, F-1 and training loss, our method respectively improved 5.13%, 5.64%, 3.60% and 5.34% compared with CNNs. Analogously, the experimental results demonstrated that compared with LSTM, our method respectively improved 3.58%, 3.03%, 3.38% and 4.63% on precision, recall, F-1 and training loss.

E. ABLATION ANALYSIS OF SEMI-SUPERVISED SCHEME AND MULTI-GRANULARITY SCHEME

In this section, we demonstrate the influence of the semi-supervised module and the multi-particle model on the proposed model by ablation experiments. We exclude semi-supervised module and multi-granularity module from SSMGCNNs one by one and report the results in TABLE 4.

From the results, we can observe that the semi-supervised module and the multi-granularity module improve the performance of the model more or less. Apparently, the semi-supervised module contributes the most to the improvement, which brings about 2%-5% increment on four metrics.

Semi-supervised module is more distinguishable than multi-granularity module in two text classification tasks: car operation command and IMDB. On the car operation command data set, the experimental results showed that in terms of precision, recall, F-1 and training loss, our scheme respectively improved 3.44%, 4.06%, 4.23%, and 2.98% compared with the model excluding semi-supervised module from SSMGCNNs. Similarly, the experimental results showed that, compared with the model excluding multi-granularity module from SSMGCNNs, our scheme respectively improved 2.18%, 1.51%, 1.01% and 4.42% on precision, recall, F-1 and training loss.

F. MODEL ANALYSIS

We also conduct experiments to compare the results using different internal structures on the real-world car operation

TABLE 3. Model performance in terms of precision, recall, F-1, and training loss.

	Car Operation Command Data Set				IMDB			
	Precision	Recall	F-1	Training Loss	Precision	Recall	F-1	Training Loss
SVM	85.38%	83.85%	82.94%	11.38%	88.94%	82.57%	85.84%	10.07%
LSTM	92.31%	89.72%	90.59%	9.02%	93.95%	90.02%	92.76%	8.73%
CNNs	90.76%	87.11%	90.37%	9.73%	92.36%	87.92%	90.31%	9.01%
Semi-supervised model	92.16%	89.42%	91.03%	9.16%	95.42%	91.13%	94.01%	7.88%
Our work	95.89%	92.75%	93.97%	4.39%	97.83%	95.57%	95.97%	4.03%

TABLE 4. Ablation analysis of semi-supervised scheme and multi-granularity scheme.

	Car Operation Command Data Set				IMDB			
	Precision	Recall	F-1	Training Loss	Precision	Recall	F-1	Training Loss
w/o semi-supervised module	92.45%	88.69%	89.74%	7.37%	94.75%	90.84%	93.22%	6.97%
w/o multi-granularity module	93.71%	91.24%	92.96%	8.81%	95.76%	92.43%	95.07%	7.12%
Our work	95.89%	92.75%	93.97%	4.39%	97.83%	95.57%	95.97%	4.03%

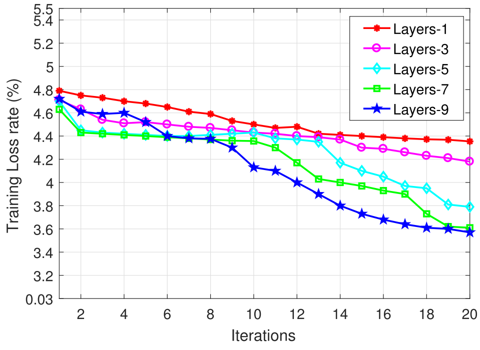


FIGURE 9. The change of training loss with the increase of the number of hidden layers.

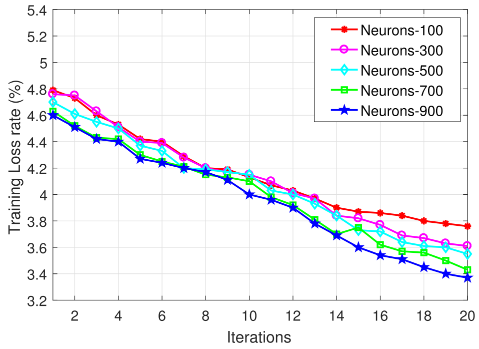


FIGURE 10. The change of training loss with the increase of the number of neurons.

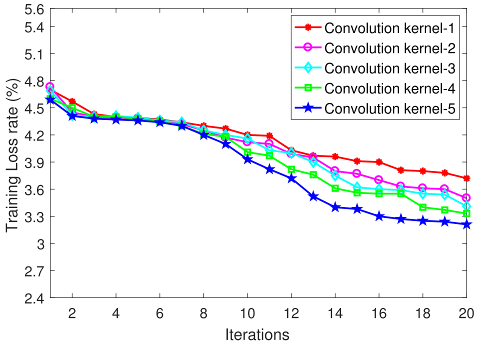


FIGURE 11. The change of training loss with the increase of the number of convolution kernels.

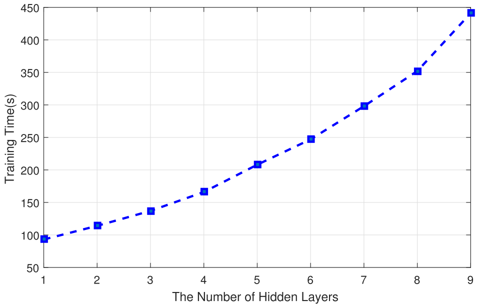


FIGURE 12. Training time of different hidden layers.

command data set. As shown in FIGURE 9, the horizontal and vertical coordinates represent the number of iteration and the training loss rate, respectively. It is found that the network depth improves model performance. The main reason is that the more hidden layers can extract more feature information to use in the top layer. However, as the number of hidden layers increases, the model becomes more and more complex, so that more time is needed for training as shown in FIGURE 12 where the horizontal and vertical coordinates represent the number of hidden layers and the training time of the model, respectively.

Moreover, we observe the influence of neuron numbers on the model performance, as shown in FIGURE 10. The results show that as the number of neurons increases, the training loss rate of the model decreases. In other words, the number of neurons improves the semantic analysis performance. Nonetheless, as the number of neurons increases, the model becomes more and more complex due to more and more internal parameters, so that more time is needed for training (cf. FIGURE 13).

Besides, we note that the number of convolution kernels (namely, region size) in the same convolution layer has an impact on the model performance, as shown in FIGURE 11. (Note that *convolution kernel* − 1 represents region size = 2; *convolution kernels* − 2 represents region size = 2 and region size = 3; *convolution kernels* − 3 represents region size = 2, region size = 3 and region size = 4; *convolution kernels* − 4 represents region size = 2, region size = 3, region size = 4 and region size = 5; and *convolution kernels* − 5 represents region size = 2, region size = 3, region size = 4, region size = 5 and region size = 6.). This indicates that feature extraction is simultaneously carried out on the same convolutional layer, thus text features of different dimensions are obtained. However, as the number of convolution kernels increases, the model becomes

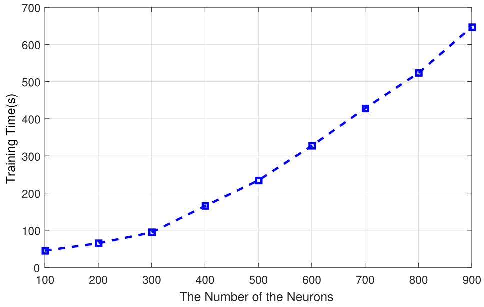


FIGURE 13. Training time of different neurons.

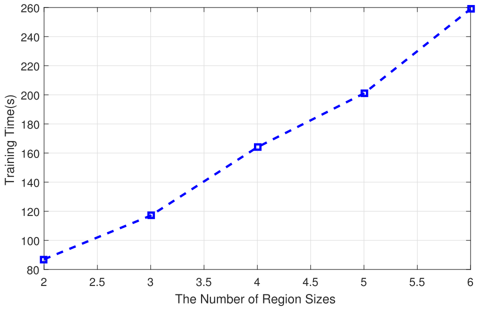
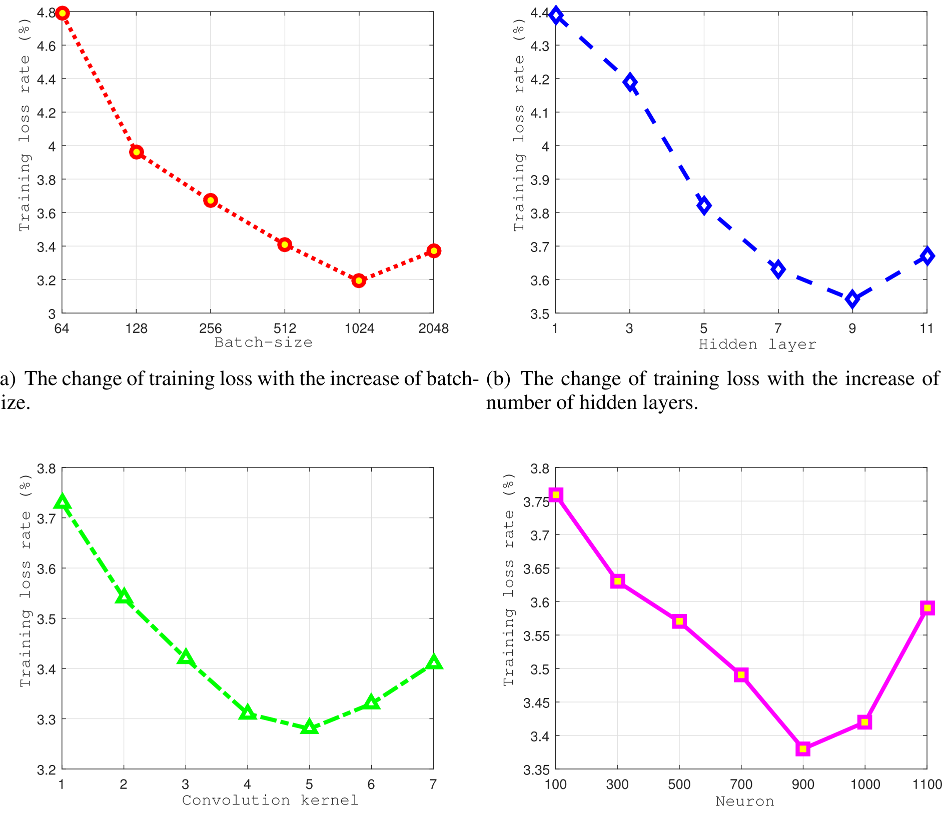


FIGURE 14. Training time of different convolution kernels.

more and more complex, so that it takes more time to train (FIGURE 14).

The whole car data set is randomly segmented into training sets (16K), validation sets (10K) and testing sets (7K). During training, weights and biases are updated; during validation, hyper-parameters are adjusted; during testing, model performance can be evaluated. In terms of hyper-parameter setting, batch size, the number of hidden layers, the number of convolution kernels and the number of neurons can be optimized manually through 60-fold cross-validation. Note that since the loss function was convergent during 75 epoches, epoches are set as 60. Keeping other parameters unchanged, SSMGCNNs runs 60 epoches setting batch size to 64, 128, 256, 512, 1024 and 2048. The average loss value of each batch size during 60 epoches is deemed as the evaluating indicator. FIGURE 15(a) shows that the loss value of batch-size = 1024 is lower than that of other batch sizes. Thus, batch size of SSMGCNNs to be optimized is set as 1024. Then, keeping batch-size = 1024 and the rest of the parameters unchanged, the number of hidden layers is set as 1, 3, 5, 7, 9 and 11. FIGURE 15(b) indicates the best setting of hidden layers is 9 under the condition of batch-size = 1024. Next, keeping batch-size = 1024, the number of hidden layers = 9 and the rest of the parameters unchanged, the number of convolution kernels is set as 1, 2, 3, 4, 5, 6 and 7. FIGURE 15(c) indicates the best setting of convolution kernels is 5 under the condition



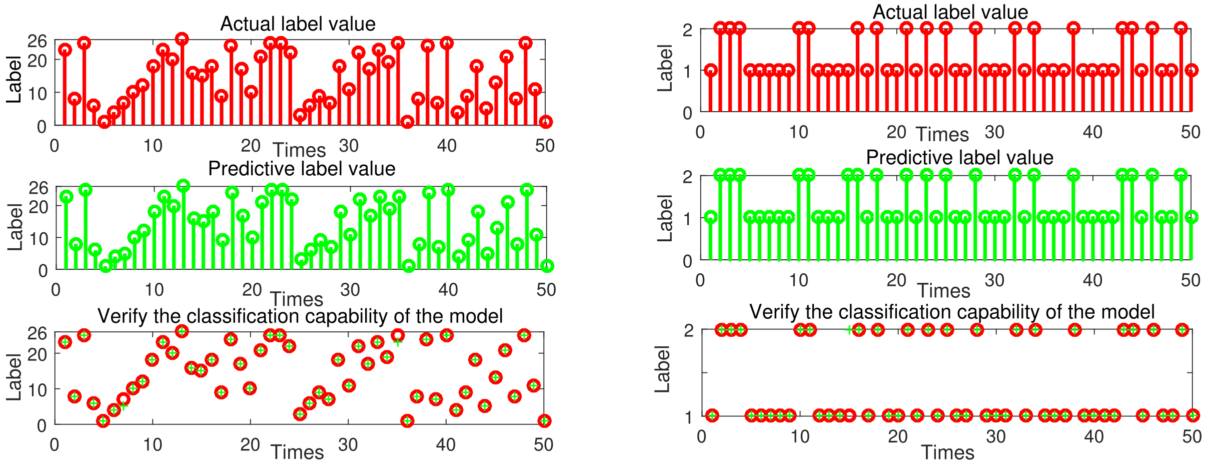
(a) The change of training loss with the increase of batch-size.

(b) The change of training loss with the increase of the number of hidden layers.

(c) The change of training loss with the increase of the number of convolution kernels.

(d) The change of training loss with the increase of the number of neurons.

FIGURE 15. Training loss rate (%) of SSMGCNNs during 60-fold cross-validation.



(a) The car operation command data set as input.

(b) The movie review data set as input.

FIGURE 16. **Testing the text classification model based on SSMGCNNs.**

of batch-size = 1024 and the number of hidden layers = 9. Finally, keeping batch-size = 1024, the number of hidden layers = 9, the number of convolution kernels = 5 and the rest of the parameters unchanged, the number of neurons is set as 100, 300, 500, 700, 900, 1000 and 1100. FIGURE 15(d) indicates the best setting of neurons is 900 under the condition of batch-size = 1024, the number of hidden layers = 9 and the number of convolution kernels = 5. In addition, dropout remains to be 0.1 because there is no overfitting phenomenon in the training process.

G. PERFORMANCE EVALUATION

The SSMGCNNs-based model is applied to text classification. From above training models, the model with the least training loss is selected to test the performance of text classification. In FIGURE 16, actual labels are represented by the red "○", and predicted labels are represented by the green "○". During verifying the classification ability, the true label is represented by the red "○", and the predicted value is represented by the green "+". Observing the label values between the red "○" and the green "+", the classification performance of the model can be obtained. FIGURE 16 shows the text classification performance of different data sets. The car operation command data set is used for model input to evaluate the classification performance of the proposed model, as shown in FIGURE 16 (a). The label of every car operation command is predicted. Analogously, the movie review data set (i.e. IMDB) is used for model input to evaluate the classification performance, as shown in FIGURE 16 (b).

H. CASE STUDIES

Text classification task is one of the pivotal problems in NLP field. DNNs have become a powerful learning machine, making it possible to work with text itself as raw input. In this section, we present the detailed case studies on text classification task.

TABLE 5. **Sample questions correctly answered via labels.**

Input: How can I get to the Forbidden City in Beijing? **Label**: 7
Input: It's raining. **Label**: 9
Input: How to get to the nearest gas station? **Label**: 7
Input: The rain has stopped. **Label**: 10
Input: It's so hot today. **Label**: 5
Input: It's so boring. **Label**: 9

TABLE 5 shows sample test questions from the car operation command data set and the label. These questions are the text of human drivers interacting with cars, and the label represents the category of every car operation command. For every pair of question and label in this table, note that each question corresponds to a label; hence it belongs to single-label classification. After the question is entered, the system outputs the corresponding label. We note that the system also outputs wrong labels in rare cases, e.g., when the input is "It's so boring.", the wrong output is "9". However, the right output should be "1" (namely "Open the music."). (Please note that refer to "Data Sets" section for specific numerical representations.)

VI. CONCLUSION

To alleviate the problem of data annotation and feature extraction during the processes of text classification, a variant of CNNs, called SSMGCNNs, is proposed in this study. SSMGCNNs consist of two parts: TVE module and MGCNNs module. TVE model learns embeddings of text regions from the unlabeled data and then integrate the learned embeddings into MGCNNs. MGCNNs can extract more feature information hidden in text by multiple convolution kernels of the same convolution layer. The system constructs the automobile semantic analysis, completes the user intention reasoning and realizes human-car interaction such as intelligent navigation, intelligent entertainment and autonomous control of car, which greatly improves user experience in the current market and enables users to enjoy high-quality auto function. Experimental results demonstrated that our method achieves better performance compared with baselines. On the car operation command data set, the experimental results demonstrated that in terms of precision, recall, F-1 and training loss, our scheme respectively improved 5.13%, 5.64%, 3.60% and 5.34% compared with CNNs. However, the extent to which this type of model can be scaled to much larger and wider domains remains an open question which we hope to pursue in our further work.

REFERENCES

[1] N. Chen, W. Liu, R. Bai, and A. Chen, "Application of computational intelligence technologies in emergency management: A literature review," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 2131–2168, Oct. 2019.

[2] L. Guo, S. Manglani, Y. Liu, and Y. Jia, "Automatic sensor correction of autonomous vehicles by human-vehicle teaching-and-learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8085–8099, Sep. 2018.

[3] Y. Zhang, G. Zhang, R. Fierro, and Y. Yang, "Force-driven traffic simulation for a future connected autonomous vehicle-enabled smart transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2221–2233, Jul. 2018.

[4] K. Huang, B. Hu, L. Chen, A. Knoll, and Z. Wang, "Adas on cots with OpenCL: A case study with lane detection," *IEEE Trans. Comput.*, vol. 67, no. 4, pp. 559–565, Apr. 2018.

[5] E. Erzin, Y. Yemez, A. M. Tekalp, A. Ercil, H. Erdogan, and H. Abut, "Multimodal person recognition for human-vehicle interaction," *IEEE MultimediaMag.*, vol. 13, no. 2, pp. 18–31, Apr. 2006.

[6] R. Tian, L. Li, V. S. Rajput, G. J. Witt, V. G. Duffy, and Y. Chen, "Study on the display positions for the haptic rotary device-based integrated in-vehicle infotainment interface," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1234–1245, Jun. 2014.

[7] M. Tanelli, R. Toledo-Moreo, and L. M. Stanley, "Guest editorialholistic approaches for human–vehicle systems: Combining models, interactions, and control," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 5, pp. 609–613, Sep. 2017.

[8] "Holistic approaches for human-vehicle systems: Combining models, interactions and control," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 3, p. 484, Jun. 2016.

[9] S. C. Lee and Y. G. Ji, "Complexity of in-vehicle controllers and their effect on task performance," *Int. J. Hum.-Comput. Interact.*, vol. 35, no. 1, pp. 65–74, Jan. 2019.

[10] C. Xu, Y. Zhang, D. Guo, W. Wang, and B. Liu, "System design of driving behavior recognition based on semi-supervised learning," in *Proc. Int. Conf. Hum. Centered Comput.* Springer, 2018, pp. 535–546.

[11] M. Li and Z.-H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 6, pp. 1088–1098, Nov. 2007.

[12] G. Tur, D. Hakkani-Tür, and R. E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," *Speech Commun.*, vol. 45, no. 2, pp. 171–186, Feb. 2005.

[13] M. Belkin and P. Niyogi, "Semi-supervised learning on Riemannian manifolds," *Mach. Learn.*, vol. 56, nos. 1–3, pp. 209–239, Jul. 2004.

[14] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Mach. Learn.*, vol. 28, no. 2, pp. 133–168, 1997.

[15] J. An, L. Fu, M. Hu, W. Chen, and J. Zhan, "A novel fuzzy-based convolutional neural network method to traffic flow prediction with uncertain traffic accident information," *IEEE Access*, vol. 7, pp. 20708–20722, 2019.

[16] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Proc. Adv Neural Inf Process Syst*, vol. 28, 2015, pp. 919–927.

[17] J. Xu and Y. Cai, "Incorporating context-relevant knowledge into convolutional neural networks for short text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 10067–10068.

[18] L. Yao, C. Mao, and Y. Luo, "Clinical text classification with rule-based features and knowledge-guided convolutional neural networks," *BMC Med. Informat. Decis. Making*, vol. 19, no. S3, p. 71, Apr. 2019.

[19] Y. He, J. Li, Y. Song, M. He, and H. Peng, "Time-evolving text classification with deep neural networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2241–2247.

[20] J. Zheng and L. Zheng, "A hybrid bidirectional recurrent convolutional neural network attention-based model for text classification," *IEEE Access*, vol. 7, pp. 106673–106685, 2019.

[21] I. Banerjee, Y. Ling, M. C. Chen, S. A. Hasan, C. P. Langlotz, N. Moradzadeh, B. Chapman, T. Amrhein, D. Mong, D. L. Rubin, O. Farri, and M. P. Lungren, "Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification," *Artif. Intell. Med.*, vol. 97, pp. 79–88, Jun. 2019.

[22] E. D. S. Maldonado, E. Shihab, and N. Tsantalis, "Using natural language processing to automatically detect self-admitted technical debt," *IEEE Trans. Softw. Eng.*, vol. 43, no. 11, pp. 1044–1062, Nov. 2017.

[23] M. Grbovic, N. Djuric, S. Guo, and S. Vucetic, "Supervised clustering of label ranking data using label preference information," *Mach. Learn.*, vol. 93, nos. 2–3, pp. 191–225, Nov. 2013.

[24] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "PCANet: A simple deep learning baseline for image classification?" *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5017–5032, Dec. 2015.

[25] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," 2014, *arXiv:1412.1058*. [Online]. Available: http://arxiv.org/abs/1412.1058

[26] A. V. Uriarte-Arcia, I. López-Yáñez, and C. Yáñez-Márquez, "One-hot vector hybrid associative classifier for medical data classification," *PLoS ONE*, vol. 9, no. 4, 2014, Art. no. e95715.

[27] S.-C.-B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural Netw.*, vol. 8, nos. 7–8, pp. 1201–1214, Jan. 1995.

[28] N. J. Bershad, M. Ibnkahla, and F. Castanie, "Statistical analysis of a two-layer backpropagation algorithm used for modeling nonlinear memoryless channels: The single neuron case," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 747–756, Mar. 1997.

[29] N. Meuleau and M. Dorigo, "Ant colony optimization and stochastic gradient descent," *Artif. Life*, vol. 8, no. 2, pp. 103–121, Apr. 2002.

[30] J. Shen, H. Xu, and P. Li, "Online optimization for max-norm regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1718–1726.

[31] G. Li, Z. Wang, and Y. Ma, "Combining domain knowledge extraction with graph long short-term memory for learning classification of Chinese legal documents," *IEEE Access*, vol. 7, pp. 139616–139627, 2019.

[32] E. Leopold and J. Kindermann, "Text categorization with support vector machines. How to represent texts in input space?" *Mach. Learn.*, vol. 46, nos. 1–3, pp. 423–444, 2002.

FEN ZHAO was born in Zaozhuang, Shandong, China, in 1991. She is currently pursuing the Ph.D. degree with the Chongqing University of Posts and Telecommunications, China. Her research interests include principle and application of neural computation natural language processing, recognition of speech signals, intelligent control theory, and pattern recognition.

YINGUO LI was born in Huangmei, Hubei, China, in 1955. He received the Ph.D. degree in control science and engineering from Chongqing University, Chongqing, in 2003. Since 2007, he has been a Professor with the Automation Department, Chongqing University of Posts and Telecommunications. His interests include computer vision on autonomous vehicle, advance driving assistant systems, intelligent control theory, and pattern recognition. He received the First and Second Prize of Chongqing Science and Technology Progress, in 2009 to 2014, respectively.

LING BAI was born in Shijiazhuang, Hebei, China, in 1991. She is currently pursuing the Ph.D. degree with the Chongqing University of Posts and Telecommunications, China. Her research interests include principle and application of neural computation, speech signals, intelligent control theory, and pattern recognition.

ZHEN TIAN received the B.S. and M.S. degrees in control science and engineering from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree in computer science and technology. His main research interests include natural language processing, intelligent control theory, and signal processing in intelligent and connected vehicle. He was funded by the China Scholarship Council as a co-training Ph.D. student to study at McMaster University, Hamilton, ON, Canada, from 2018 to 2019.

XINHENG WANG was born in Zaozhuang, Shandong, China, in 1988. He is currently pursuing the Ph.D. degree with the Chongqing University of Posts and Telecommunications, China. His research interests include natural language processing, recognition of speech signals, and intelligent control theory.