

## Research Article

# Mapping Relationship Discovery of Multidimensional Architectures in Autonomous Transportation System Based on Text-Matching Model

You Liu , Haonan Tuo , Mingfan He , Qiang Fu , Tianjian Yu , and Jinjun Tang 

Smart Transport Key Laboratory of Hunan Province, School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China

Correspondence should be addressed to Tianjian Yu; [yutianjian@csu.edu.cn](mailto:yutianjian@csu.edu.cn) and Jinjun Tang; [jinjuntang@csu.edu.cn](mailto:jinjuntang@csu.edu.cn)

Received 20 June 2022; Revised 27 July 2022; Accepted 6 April 2023; Published 30 April 2023

Academic Editor: Linlin You

Copyright © 2023 You Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mapping relationships of multidimensional architectures play an essential role in the autonomous transportation system (ATS), which can help understand the complex relationships between multidimensional architectures. The current mapping relationship discovery for multidimensional architectures in ATS requires significant manual involvement, leading to the underutilization of textual data and the intense subjectivity of results. In order to address the above issues, it is necessary to mine and further utilize the semantic information in the textual data. This study applies the text-matching model to the mapping relationship discovery of multi-dimensional architectures, which can calculate the semantic similarity between texts. On this basis, a method based on the Siamese-BERT-wwm-Bi-LSTM model is proposed, which incorporates Chinese BERT with whole word masking (BERT-wwm), bidirectional long-short term memory (Bi-LSTM), and the Siamese Network. A series of experiments are conducted with different text-matching models. The results show that the precision rate, recall rate, and F1-score exceed 80% for most applied methods, which verifies the feasibility of using the text-matching model for mapping relationship discovery. These results expect to provide a method with good performance that can automatically perform mapping relationship discovery.

## 1. Introduction

With the development of emerging technologies, such as artificial intelligence and 5G communication, the ability of transportation systems to self-organize operating and autonomous serving is rapidly improving [1–3]. Over time, the intelligent transportation system (ITS) is gradually transforming into a proactive Autonomous Transportation System (ATS), which is a new generation of systems with less human intervention and greater autonomy [4–6]. Compared to ITS supported by packet-switched networks, ATS supported by emerging technologies can automatically and intelligently manage mobility demand and supply through a self-actuating cycle consisting of sensing, learning, rearranging, and reacting steps [7]. For the new transportation system, a matching transportation system framework needs

to be proposed. China, the United States, and Europe have recently conducted many studies on the transportation system framework. However, these studies generally focus on urban transportation system services, resulting in weak adaptability to emerging technologies. Therefore, there is a need to construct ATS architecture foundation theories capable of studying road traffic and integrated transportation in depth [8–13]. The multidimensional architectures in ATS consist of functional, logical, and physical architectures, which describe the system components from the corresponding perspectives. The mapping relationships of multidimensional architectures reveal the matching relationship between the constituent elements of different architectures. The in-depth elaboration of mapping relationships is one of the essential aspects of the theoretical study for ATS architectures, which helps to provide

academic support for the complex relationship exploration and the generational evolution study of multidimensional architectures.

While constructing the multidimensional architectures in ATS, textual data such as function sets, data dictionaries, physical entity object tables, and knowledge bases in different service scenarios are formed. These textual data contain constituent elements belonging to different architectures and information flow in various service scenarios. The textual data provides essential references for the mapping relationship discovery of multidimensional architectures [14]. Methods based on the fusion of horizontal and vertical and the fuzzy theory have already been applied to study ATS architectures. These two approaches alleviate the intense subjectivity and underutilization of textual data by establishing specific mapping relationship construction principles [15–18]. However, the methods above still suffer from the high requirement for manual involvement and the difficulty of directly utilizing textual data.

The text-matching model-based method is a prominent approach to studying the mapping relationship through big data mining and textual information extraction. The method determines whether two texts have a matching relationship based on constructing a model to extract their linguistic features. Several text-matching approaches, such as word statistical information-based, grammatical structure analysis-based, and semantic information-based, have been applied to explore the mapping relationship. However, the application of the first two methods is often limited by their shortcomings. Precisely, the method based on word statistical information can only extract shallow features such as word frequency while ignoring deep features such as the text structure and semantic information [19–21]. The methods of grammatical structure analysis require plenty of textual data with part-of-speech annotations, which have terrible performance in dealing with the domain text with many specialized words. In contrast, methods based on semantic information have recently been widely used in text matching. These methods can mine the deep semantic information of the textual data and support the text-matching decision by introducing the deep learning model based on the word embedding models.

The neural-network models, such as convolution neural network (CNN), recurrent neural network (RNN), and long-short term memory (LSTM), have been widely applied for deep learning. CNN can efficiently extract local features, the linguistic features at different granularities, of the textual data [22]. Yin et al. [23] introduced the attention mechanism to CNN and developed an attention-based CNN (ABCNN) for text matching. However, CNN performs poorly in extracting serialization features and long-distance dependencies from the textual data. Conversely, RNN can better extract serialization features by retaining historical information, but the vanishing and exploding gradient problem of RNN lead to learning only short-term information in the network [24]. LSTM has solved the problem of learning long-term information on account of the application of the gating mechanism. Generally, LSTM includes two main approaches: one-way LSTM and bidirectional long-short term memory

(Bi-LSTM). One-way LSTM tends to neglect the beginning part of the sequences while processing long sequences [25]. Moreover, it cannot learn the information from subsequent units, which makes it challenging to handle specific problems that require the fusion of backward and forward information. In contrast, Bi-LSTM is more capable of processing textual data with an emphasis on order, and it can learn the bidirectional information by combining the forward LSTM and backward LSTM. Palangi et al. [26] introduced Siamese-Bi-LSTM for text-matching based on combining Siamese Network and Bi-LSTM. Chen et al. [27] presented the text-matching model, Enhanced LSTM for natural language inference (ESIM), by integrating the attention mechanism and Bi-LSTM.

Previous findings illustrated that the word embedding model is the other meaningful part of the text-matching model based on semantic information. It can represent text in a low-dimensional space by converting the words into dense numeric vectors that the computer can process. Mikolov et al. [28] presented the word representation model, Word2vec, by applying the continuous bag-of-words model and the continuous skip-gram model. Pennington et al. [29] created a global word representation model, GloVe, based on the co-occurrence matrix of words from the corpus. Both Word2vec and GloVe are static word embedding models, which are disabled to solve the problems of semantic nesting and polysemy. The extractor for deep bidirectional text features, bidirectional representation from transformers (BERT), solved the problem remarkably [30]. With the masked language model (MLM) and next sentence prediction (NSP) as pretraining tasks, BERT can extract deep semantic information from textual data. Moreover, by adjusting the details of BERT, it can generate a series of derived models. Liu et al. [31] discovered a robustly optimized BERT pretraining approach (Roberta), which uses a dynamic masking mechanism and removes the next sentence prediction mission. Lan et al. [32] presented a lite BERT (ALBERT) by reducing the quantities of parameters. Cui et al. [33] recommended the Chinese BERT with whole word masking (BERT-wwm), which was more suitable for processing Chinese textual data. Reimers and Gurevych [34] put forward the sentence-BERT with pretraining in Siamese network. Compared to BERT, Sentence-BERT can represent sentences better as embedding vectors, making it feasible to evaluate the text match based on the cosine similarity of sentence embedding vectors. In this way, Sentence-BERT can solve the problem of taking too many computer sources and the slow prediction of applying BERT for text-matching. It should be noted that the outputs of BERT and its derivative models are complex and need dimensionality reduction. The common pooling methods for dimensional reduction, such as [CLS] pooling and mean pooling, tend to result in partial information loss. Dong et al. [35] suggested combining BERT and CNN, extracting more detailed semantic information. Agrawal et al. [36] replaced pooling operations with Bi-LSTM, representing sentences better in vector space. Choi et al. [37] and Viji and Revathy [38] applied the Siamese Network for text-matching based on BERT-CNN and BERT-Bi-LSTM, achieving high matching accuracy.

This paper aims to develop a hybrid method, Siamese-BERT-wwm-Bi-LSTM, simultaneously considering engineering efficiency and matching accuracy. Siamese-BERT-wwm-Bi-LSTM takes the Siamese Network as the basic structure in which the combination of the superior Chinese pretraining model BERT-wwm and Bi-LSTM can accurately calculate the semantic similarity of text. The proposed method can perform mapping relationship discovery automatically by learning the semantic information in textual data, which alleviates the underutilization of textual data, low efficiency, and high subjectivity of the existing methods.

The remainder of this paper is organized as follows. Section 2 introduces the framework and methodology of this study. In Section 3, data sources and experiment details are presented. In Section 4, we discussed the performance of various models by analyzing the results. In Section 5, we summarize the conclusion of this paper.

## 2. Methodology

The multidimensional architectures of ATS consist of functional, logical, and physical architectures, whose essence is to describe the composition of ATS from three perspectives. The mapping relationships illustrate matching relationships between constituent elements from distinct architectures. Figure 1 shows an example of the mapping relationship of elements for monitoring traffic events. In different architectures, the same traffic scenario will be parsed into the various elements of the different architectures. In the functional architecture, monitoring traffic events is resolved into functional elements, such as collecting traffic information, generating traffic event information, and posting traffic event information. This same scene is decomposed into logical elements in the logical architecture, such as network information, traffic status information, and location and traffic events. Physical elements, such as roadside devices, network management platforms, and traffic operations managers, achieved monitoring traffic events in the physical architecture. Among them, collecting traffic information is implemented in reality by passing the location of traffic events from roadside devices. These three elements from different architectures form a complete data flow, as shown in Table 1. The architecture elements involved in the same data flow are identified as having mapping relationships with each other. The mapping relationship is shown in Table 2, where the label “1” means a mapping relationship between the front two elements. In the previous studies, the mapping relationship discovery of architecture elements was achieved by manual analysis of elements. However, manual analysis is always with the problem of inefficiency and underutilization of textual data. For the issues above, this study applies a method based on semantic information to discover mapping relationships automatically. The name of elements contains rich information, and the names of elements with mapping relationships have similar semantic information. It can be determined if there is a mapping relationship between elements by analyzing their semantic similarities. Additionally, the development of NLP brings the ability to mine and extract the reliable semantic

information of elements. Therefore, it is a feasible way to discover mapping relationships by text-matching based on semantic analysis.

This study proposes a text-matching model-based method, Siamese-BERT-wwm-Bi-LSTM, for discovering mapping relationships. The process adopts the Siamese Network as the basic framework. Based on the framework, the technique combines the superior Chinese pretraining models BERT-wwm and Bi-LSTM. Figure 2 displays the framework of Siamese-BERT-wwm-Bi-LSTM. Firstly, convert the multi-dimensional architecture data into the required mapping relationship data in Table 2. Then, input the mapping relationship data into BERT-wwm. The BERT-wwm encodes the data set into embedding vectors. Afterwards, the generated vectors are used as the input of Bi-LSTM. Bi-LSTM can extract the bidirectional features of each input data and combine the forward and reverse LSTM to get a complete hidden state sequence as the element embedding vectors. Finally, determine whether there is a mapping relationship between elements by calculating the cosine similarity of the corresponding embedding vectors. If the cosine similarity between the two is greater than or equal to 0.5, it is believed that there is a mapping relationship between them.

*2.1. Preprocessing for Textual Data.* Textual data from multidimensional architecture in ATS cannot be directly used for discovering mapping relationships. After some preprocessing, these data need to convert to the form shown in Table 2 above after some preprocessing. Firstly, extract the data flow like Table 1 from the multidimensional architecture data. It is supposed that different elements involved in the same data flow have a mapping relationship. According to the principle, withdraw the mapping relationship data from different data flows. It is difficult for the model to learn the boundaries of the mapping relationship from only positive samples. Therefore, expand the data by using random negative sampling to enhance its feature. Randomly select some pairs of elements that do not have a mapping relationship with each other to expand the data. The expanded data is shown in Table 3. In Table 3, the label “1” respectively marks a mapping relationship between the front two elements, and “0” represents the opposite.

*2.2. Pretraining Model: BERT-wwm.* When processing textual data, the small size of the labeled example makes it hard for the model to learn enough information. Pretraining models solve the issue above. Firstly, the pretraining model forms through self-supervised learning from large-scale unlabeled data. The developed model can extract a word’s semantic representation in a particular context. Secondly, finetune the model according to the specific tasks, which means tuning the parameters of the pretraining based on the labeled samples. In recent years, the approach based on pretraining models has become common in NLP.

BERT and its derivative models are the most popular pretraining models in recent years. The textual data of multidimensional architectures are in Chinese. Hence, the

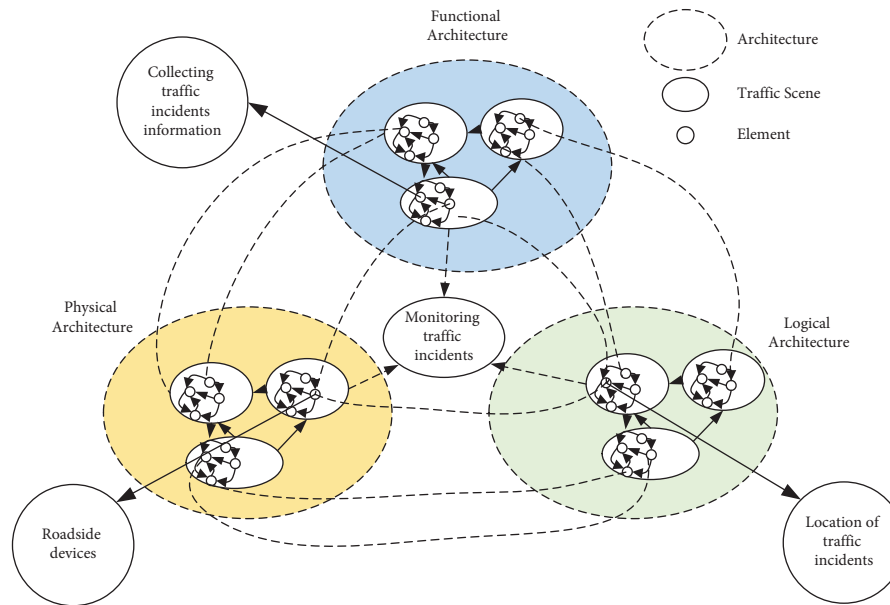


FIGURE 1: The mapping relationship of elements in monitoring traffic incidents.

pretraining models involved in this study are all Chinese versions. Figure 3 displays the structure of BERT-wwm, which is the same as the official Chinese BERT. The network architecture uses a multilayer transformer structure [33], which transforms the distance between two words at any position into “1” through an attention mechanism, effectively solving the thorny long-term dependency problem in NLP [39]. Figure 4 is the diagram of a transformer block corresponding to a “Trm” in Figure 3. The transformer adopts a multi-headed self-attention mechanism that enables it to learn hidden information from distinct semantic scenarios. BERT-wwm is based on the bidirectional transformer that jointly adjusts the left-to-right and right-to-left transformers. This approach effectively considers the interactivity between words. Figure 4 demonstrates the framework of the transformer block. By processing  $N$  combinations of multiheaded self-attention mechanisms and feedforward fully connected networks, the word in the sentence converts into an abstract representation of embedding vectors. The vectors generated in this way contain richer semantic information. Figure 5 shows the input for BERT-wwm. The model input forms by splicing word, segment, and position embeddings. Taking the sentence “Intelligent Transportation System is a hotspot” as an example, the sentence is shredded into separate words with [CLS] and [SEP] identified and added. [CLS] identifies the start of a sentence, and [SEP] is the symbol for separating two independent sentences.

The differences between the official Chinese BERT and BERT-wwm are mainly pretraining tasks. The pretraining tasks for BERT are MLM and NSP. For the MLM task, the training approach is to randomly replace the tokens in the input text sequence with the identifier [MASK] and then predict those masked tokens. However, BERT’s original word segmentation method based on Word Piece [30] will cut a complete term into several tokens. This approach can extract

only token-level semantic features without considering the traditional Chinese word segmentation. Whole word masking (WWM) will mask the other parts of the same term if a part of a complete term is masked. In this way, the term-level semantic information can be extracted. While remaining the NSP task, BERT-wwm can better deal with Chinese NLP issues by replacing the pretraining mission MML with WWM.

**2.3. Bi-LSTM.** According to Figure 3, the output of BERT-wwm consists of embedding vectors of [CLS], [SEP], and other tokens. Due to its high dimensionality, the output needs to be processed through pooling operations in the sentence-level task. Typically, the [CLS] pooling strategy is used for dimensionality reduction, which refers to taking the [CLS] identifier vector as the whole sentence representation [28]. Compared to CLS pooling, the model’s performance can be further improved by processing the model via models such as CNN and Bi-LSTM [35–38]. Bi-LSTM, which has an excellent performance in extracting the textual sequence features, is used in this study.

Bi-LSTM is a combination of forward and backward LSTM, which is often used to model contextual information for NLP tasks. LSTM, as a variant of the RNN network, solves the problem of gradient disappearance and explosion by setting multiple gating units to achieve long-term memory. The unit control for LSTM is accomplished by an input gate, a forgetting gate, and an output gate. Figure 6 is the diagram of the structure for an LSTM unit. In Figure 6,  $x_t$  represents the input sequence.  $h_t$  is the implied layer output, i.e., the output result of each LSTM cell.  $f_t$ ,  $i_t$ ,  $O_t$ , and  $\hat{C}_t$  are respectively the forgotten gate, input gate, output gate, and candidate memory.  $\sigma$  and  $\tanh$ , respectively, represent the activation function of Sigmoid and hyperbolic tangent.  $W$  is the weight matrix. The symbol  $\times$  stands for the dot product.

TABLE 1: Data flow of multidimensional architecture.

Traffic scene	Data flow	Type	Data source	Type	Data destination	Type
Monitoring traffic events	Location of traffic events	Logical	Roadside devices	Physical	Collecting traffic events information	Functional

TABLE 2: Mapping relationship data for monitoring traffic events.

Element $p$	Element $q$	Label
Location of traffic events	Roadside devices	1
Location of traffic events	Collecting traffic event information	1
Roadside devices	Collecting traffic event information	1

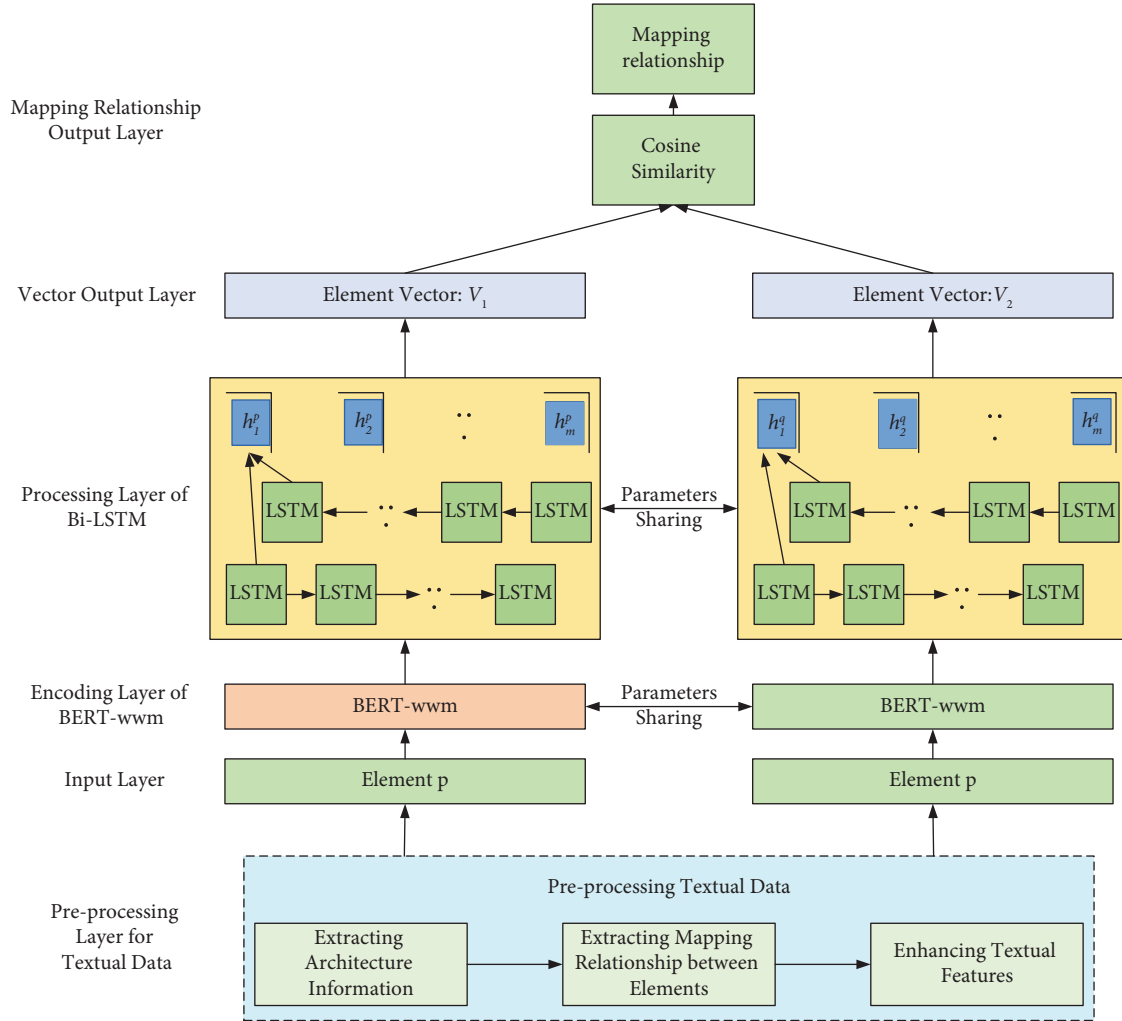


FIGURE 2: The framework of Siamese-BERT-wwm-Bi-LSTM.

TABLE 3: Expanded mapping relationship data for monitoring traffic events.

Element $p$	Element $q$	Label
Location of traffic events	Roadside devices	1
Location of traffic events	Collecting traffic event information	1
Roadside devices	Collecting traffic event information	1
Service facility information	Station management facilities	1
Service facility information	Analyzing toll station data	1
Station management facilities	Analyzing toll station data	1
Location of traffic events	Station management facilities	0
Vehicle status data	Analyzing toll station data	0

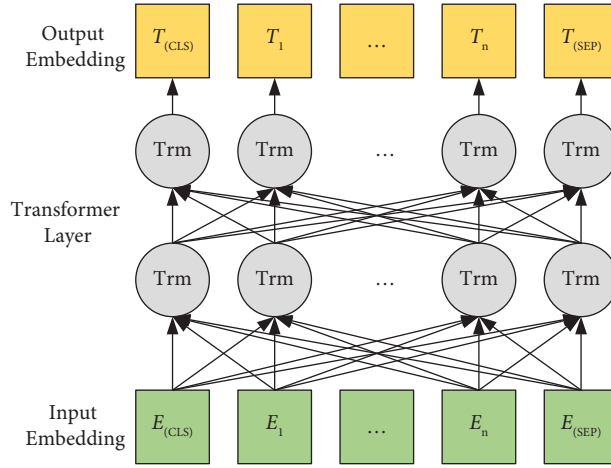


FIGURE 3: The structure of BERT-wwm.

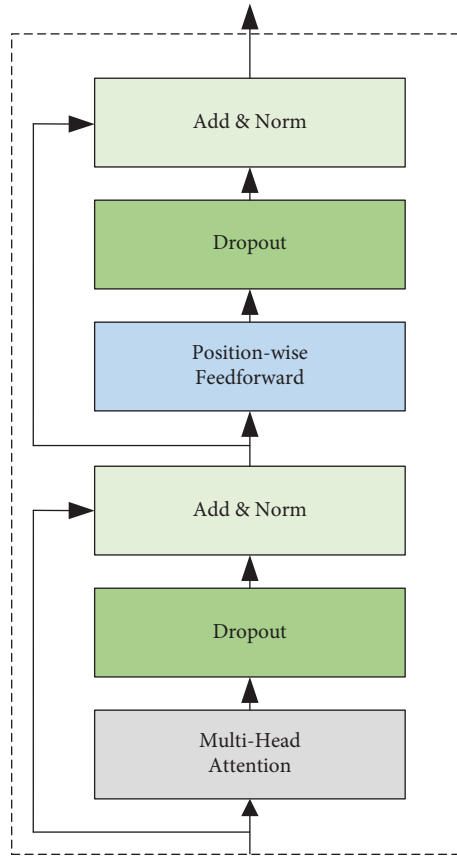


FIGURE 4: Transformer block.

The forgotten gate determines the degree to which the unit state at the last time is retained in the current state, which is calculated as follows:

$$f_t = \sigma(\mathbf{W}_f \cdot [h_{t-1}, x_t] + b_f), \quad (1)$$

where  $\mathbf{W}_f$  and  $b_f$  are the weight matrix and the deviation of the forgotten gate.

The input gate determines the number of the input to the network saved to the unit state at the current moment. It is calculated as follows:

$$\begin{aligned} i_t &= \sigma(\mathbf{W}_i \cdot [h_{t-1}, x_t] + b_i), \\ \hat{C}_t &= \tan h(\mathbf{W}_c \cdot [h_{t-1}, x_t] + b_c), \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t. \end{aligned} \quad (2)$$

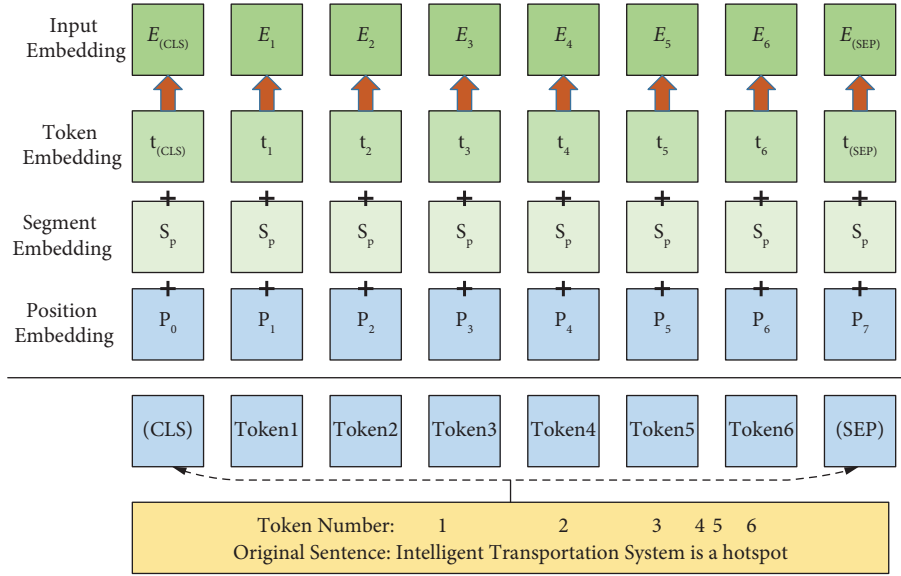


FIGURE 5: BERT-wwm input for the term intelligent traffic system.

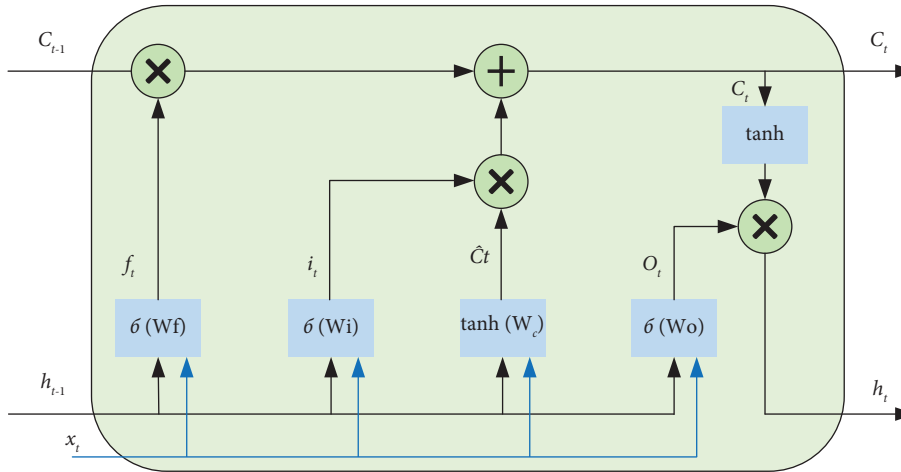


FIGURE 6: The structure of the LSTM unit.

$W_i$  and  $W_c$  are the weight matrix of the input gate and the candidate memory.  $b_i$  and  $b_c$  represent the deviation of the input gate and the candidate memory.  $\hat{C}_t$  is the candidate memory, and  $C_t$  is the current unit state.

The output gates control the number of unit states saved to the LSTM output at the current moment, which is calculated as follows:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (3)$$

$$h_t = O_t \cdot \tanh(C_t),$$

where  $W_o$  and  $b_o$  are the weight matrix and the deviation of the output gate.

Bi-LSTM learns information from previous and future moments through forward and backward LSTM. Figure 7 shows the structure of Bi-LSTM, which combines the forward and backward calculations. This approach solved the problem that only a one-way timing sequence was involved and that parameter calculations were greatly affected by the

timing sequence. In the LSTM hidden layer, the forward calculated  $\vec{h}$  and backward calculated  $\overleftarrow{h}$  are preserved. The merger of  $\vec{h}$  and  $\overleftarrow{h}$  forms the final output of Bi-LSTM.

**2.4. Siamese Network.** The Siamese network is a coupled structure built on two neural networks with mutually shared weights. While text matching, the Siamese network is commonly used as the underlying framework. The framework takes two samples as input and outputs embedding vectors as their representation in high-dimensional space to compare the similarity of the two samples. ABCNN, Siamese-Bi-LSTM, and ESIM are text-matching models using the Siamese network as a framework.

Currently, most text-matching methods based on BERT and its derivative models use the original framework recommended by the author of BERT. Figure 8 is the diagram of the framework. The process splices two sentences and inputs them into the model for semantic interaction, which takes up



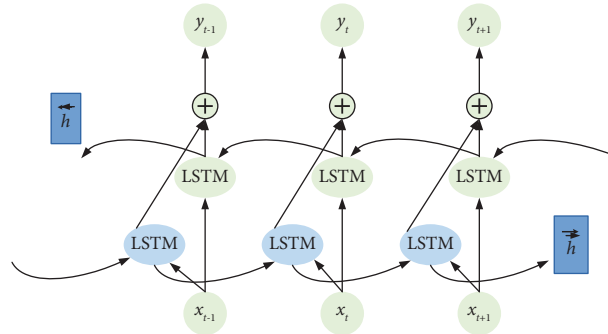


FIGURE 7: The structure of Bi-LSTM.

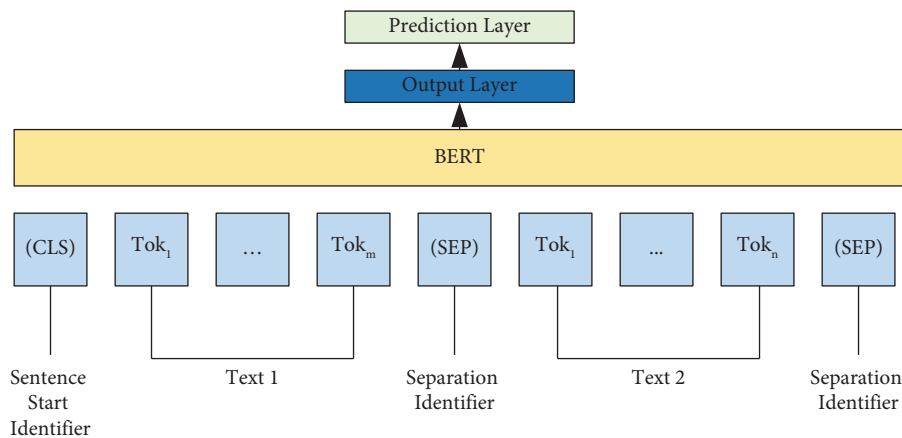


FIGURE 8: Text matching based on BERT and its derivative models.

a lot of computational resources. It requires a lot of time for performing semantic searches. Sentence-BERT adopts the structure of the Siamese Network, which can substantially reduce the time needed for semantic searches while ensuring high accuracy [34]. As shown in Figure 9, Sentence-BERT inputs two sentences independently, which can preserve the independent features of the sentences. The semantic interaction is implemented by similarity calculation. The model can represent textual data as embedding vectors with semantic information in this approach. Text matching can be achieved by calculating embedding vectors' cosine similarity and Manhattan distance, reducing computational resource consumption [34].

When performing mapping relationship discovery, it requires searching for pairs with mapping relationships among many architectural elements. The improvement of engineering efficiency is of great importance for this task. Therefore, in the hybrid model of this study, the Siamese network is used as the basic framework.

**2.5. Siamese-BERT-wwm-Bi-LSTM.** Compared with BERT and its other variant models, BERT-wwm considers the features of Chinese word segmentation. Bi-LSTM can capture more semantic information than CLS pooling. The Siamese network can improve engineering efficiency while ensuring high accuracy. This study applies a hybrid method of Siamese-BERT-wwm-Bi-LSTM for mapping relationship

discovery of the multidimensional architecture, integrating the advantages of each model. Figure 2 is the diagram for the structure of Siamese-BERT-wwm-Bi-LSTM.

### 3. Data Source and Experiment Details

**3.1. Data Set Preprocessing.** Multidimensional architecture data is generated along with the construction of each architecture. However, textual data of multidimensional architecture needs preprocessing, as shown in Section 2.1, before it can be used for mapping relationship discovery. The preprocessed data consists of 5149 items with an approximate positive to negative sample ratio of 1:1. They are divided into a training set, a validation set, and a test set in the percentage of 6:2:2. The data format is shown in Table 3.

**3.2. Experiment Environment.** The experimental hardware configuration includes a 5-core Intel (R) Xeon (R) Silver 4210R CPU 64G and an Nvidia RTX 3090 GPU 24G. The operating system is Ubuntu 18.04. The development language is Python 3.8, and the development framework is Tensorflow 2.3.0 deep learning framework.

**3.3. Experiment Setting.** To verify the effectiveness of the proposed method in this study, the following experiments are designed:

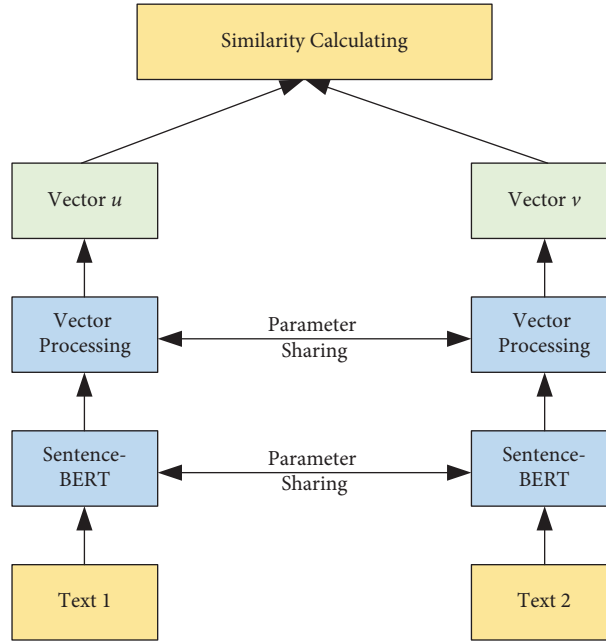


FIGURE 9: Text matching based on sentence-BERT.

- (1) Comparison of the performance of different text-matching models for discovering mapping relationships.
- (2) Ablation experiments on the hybrid model.
- (3) Comparison of the performance of BERT and its derivative models based on the structure of the proposed method.
- (4) Exploration experiments on the performance of the joint of CNN and Bi-LSTM.

Table 4 shows the parameters of the different pretraining models. The parameters are consistent with the original paper [30–33]. Table 5 shows the experimental parameter settings of the comparison models in this study. In Table 5, all the CNN-based models use Relu as the activation function. The size of the convolutional kernel in ABCNN is three. The rest of the CNN-based models use three kernels whose size is three, four, and five. In this way, the model outputs forms by splicing the features extracted from 3 kernels. F stands for the forward LSTM, and B represents the backward LSTM. CLS is on behalf of the strategy of CLS pooling. Bi-LSTM-CNN represents the Bi-LSTM output accessed into CNN as input. CNN-Bi-LSTM means the CNN output is accessed into Bi-LSTM as input. CNN + Bi-LSTM refers to the fact that the CNN output is spliced with Bi-LSTM output. Figures 10(a)–10(c) show the structure for Bi-LSTM-CNN, CNN-Bi-LSTM, and CNN + Bi-LSTM.

**3.4. Evaluation Metrics.** This study uses a text-matching model for mapping relationship discovery, transforming the problem into a binary classification. According to the session specification evaluation guidelines from the message understanding conference, apply precision rate, recall rate,

TABLE 4: Parameters of BERT and its derivative models.

Pretraining model	Number of hidden layers	Size of hidden layer	Size of output
BERT	12	768	768
ALBERT	12	768	128
Roberta	12	768	768
BERT-wwm	12	768	768

and  $F_1$ -Score as the evaluation metrics. They are calculated as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%,$$

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\%, \quad (4)$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

TP denotes the number of positive samples predicted to be positive. FP refers to the number of negative samples predicted to be positive. FN stands for the number of positive samples predicted to be negative.

## 4. Experiment Results and Analysis

Table 6 shows the results of different text-matching models in the mapping relationship discovery task, where model 9 is the developed one. The developed model achieves the best performance in evaluation metrics of precision rate, recall rate, and  $F_1$ -score for discovering mapping relationships. Compared with traditional models based on the neural network, such as model 1, model 2, and model 3, the  $F_1$ -score of model 9 is remarkably higher by more than 12%. The

TABLE 5: Parameters of text-matching model.

Text-matching model	CNN		Number of LSTM units	Dropout rate	Optimizer	Number of epochs	Batch size
	Number of filters	Size of kernels					
ABCNN	50	3	—	0.3	Adam ( $1e-3$ )	50	200
Siamese-Bi-LSTM	—	—	F: 100 B: 100	0.2	Adam ( $5e-4$ )	100	256
ESIM	—	—	F: 256 B: 256	0.3	Adam ( $5e-4$ )	50	50
BERT-wwm-CLS	—	—	—	0.1	Adam ( $2e-5$ )	20	32
BERT-wwm-Bi-LSTM	—	—	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32
BERT-wwm-CNN	128	3,4,5	—	0.1	Adam ( $2e-5$ )	20	32
Siamese-BERT-wwm-CLS	—	—	—	0.1	Adam ( $2e-5$ )	20	32
Siamese-BERT-wwm-CNN	128	3,4,5	—	0.1	Adam ( $2e-5$ )	20	32
Siamese-BERT-wwm-Bi-LSTM-CNN	128	3,4,5	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32
Siamese-BERT-wwm-CNN-Bi-LSTM	128	3,4,5	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32
Siamese-BERT-wwm-CNN + Bi-LSTM	128	3,4,5	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32
Siamese-BERT-Bi-LSTM	—	—	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32
Siamese-ALBERT-Bi-LSTM	—	—	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32
Siamese-roberta-Bi-LSTM	—	—	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32
Siamese-BERT-wwm-Bi-LSTM	—	—	F: 100 B: 100	0.1	Adam ( $2e-5$ )	20	32

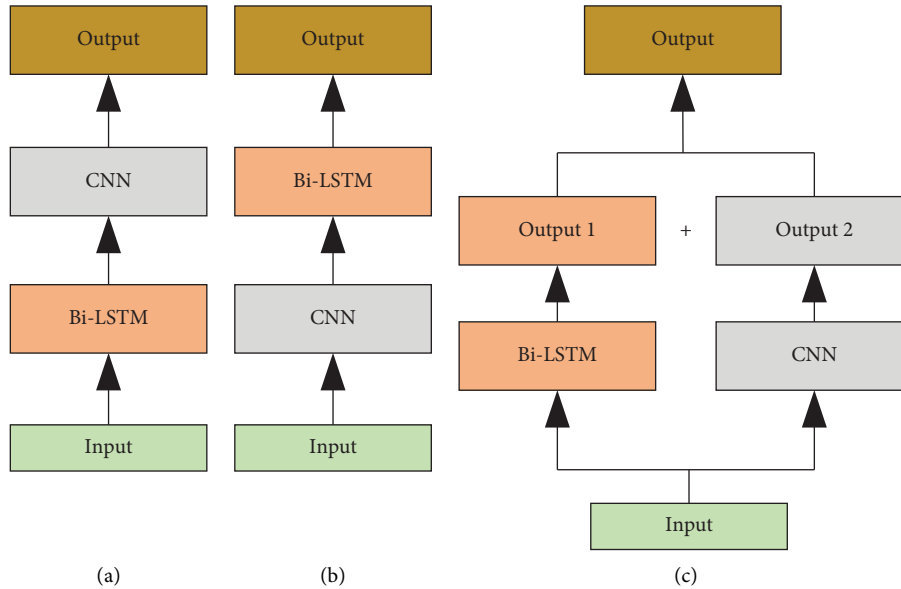


FIGURE 10: The joint use of Bi-LSTM and CNN.

results show that the pretraining model can effectively represent the textual data. Model 4, model 5, and model 6 are BERT-wwm-based models with the original framework, as

shown in Figure 8. Model 9 has an excellent  $F_1$ -score, higher than models 4, 5, and 6 by 2.69%, 4.06%, and 3.63%. It is easy to conclude that the Siamese Network can help improve the

TABLE 6: The performance of text-matching models for discovering mapping relationship.

No	Text-matching model	Precision (%)	Recall (%)	$F_1$ (%)
1	ABCNN	66.01	79.81	72.26
2	Siamese-Bi-LSTM	61.31	82.42	70.31
3	ESIM	71.13	79.57	75.11
4	BERT-wwm-CLS	84.84	85.42	85.13
5	BERT-wwm-Bi-LSTM	83.49	84.03	83.76
6	BERT-wwm-CNN	83.84	84.55	84.19
7	Siamese-BERT-wwm-CLS	85.45	85.91	85.68
8	Siamese-BERT-wwm-CNN	86.65	86.18	86.41
9	Siamese-BERT-wwm-Bi-LSTM	<b>87.30</b>	<b>88.35</b>	<b>87.82</b>

Bold values refer to the maximum prediction performance in each circumstance.

TABLE 7: The results of the ablation experiments.

No	Text-matching model	Precision (%)	Recall (%)	$F_1$ (%)
1	Siamese-Bi-LSTM	62.64	80.05	70.28
2	BERT-wwm-CLS	84.84	85.42	85.13
3	BERT-wwm-Bi-LSTM	83.49	84.03	83.76
4	Siamese-BERT-wwm-CLS	85.45	85.91	85.68
5	Siamese-BERT-wwm-Bi-LSTM	<b>87.30</b>	<b>88.35</b>	<b>87.82</b>

Bold values refer to the maximum prediction performance in each circumstance.

TABLE 8: The performance of different pretraining models.

No	Text-matching model	Precision (%)	Recall (%)	$F_1$ (%)
1	Siamese-BERT-Bi-LSTM	87.03	87.26	87.15
2	Siamese-ALBERT-Bi-LSTM	87.28	86.18	86.73
3	Siamese-roberta-Bi-LSTM	87.02	86.72	86.87
4	Siamese-BERT-wwm-Bi-LSTM	<b>87.30</b>	<b>88.35</b>	<b>87.82</b>

Bold values refer to the maximum prediction performance in each circumstance.

performance of models. Model 7, model 8, and model 9, respectively, apply the strategies of CLS pooling, CNN, and Bi-LSTM for dimensionality reduction. Their performance indicates that the strategy of processing through neural networks can help mine more detailed semantic information than CLS pooling. At the same time, Bi-LSTM shows superiority over CNN in semantics mining.

Table 7 displays the results of the ablation experiments, where model 5 is the proposed one. By comparing the results in Table 7, BERT-wwm, Bi-LSTM, and Siamese Network have improved the model performance. The metrics of model 1 and model 5 demonstrate the usefulness of the pretraining model in the mapping relationship discovery task. It can be proved that replacing CLS pooling with Bi-LSTM from the performance of model 2, model 3, model 4, and model 5. The comparison between model 4 and model 5 indicates the Siamese network is positive for the performance of models. While setting the number of epochs as 20, model 3 takes 7726 s to train based on the training set and 774 s to predict based on the validation set. With the same number of epochs, model 5 takes 3928 s to train based on the training set and 75 s to predict based on the validation set. The differences between the time lengths for training and prediction indicate that the Siamese network can effectively reduce the consumption of computing resources. In general, the various modules of the hybrid model are valid. And the Siamese network can improve engineering efficiency.

Table 8 shows the results of replacing BERT-wwm with other pretraining models in the introduced framework. The presented framework is shown in Figure 2. The pretraining models involved are BERT, Albert, and Roberta. The abovementioned models are all pretraining based on the Chinese Wikipedia Corpus. According to the metrics in Table 8, BERT-wwm shows superiority over other pretraining models. The results indicate that the mission of WWM can effectively improve the performance of pretraining models in dealing with Chinese NLP problems.

The performance of the models in Table 8 varies widely in the value of the precision rate and narrowly in the value of the recall rate. It suggests that BERT-wwm can help identify more mapping relationships while ensuring the same precision rate.

In some NLP scenarios, the joint use of Bi-LSTM and CNN works better than either model alone [40–43]. No study attempts to jointly use LSTM and CNN for text-matching models based on BERT and its derivative models. Therefore, the experiments in Table 9 are conducted, where model 5 is the proposed one. Model 1 forms by replacing Bi-LSTM with Bi-LSTM-CNN in the developed framework, as shown in Figure 2. Bi-LSTM-CNN first extracts long-range features from the output of the hidden layer by Bi-LSTM and then inputs the long-range features into CNN for the extraction of local features [40]. The Bi-LSTM part of the developed framework is changed

TABLE 9: The performance for the combination of Bi-LSTM and CNN.

No	Text-matching model	Precision (%)	Recall (%)	$F_1$ (%)
1	Siamese-BERT-wwm-Bi-LSTM-CNN	86.37	85.37	85.87
2	Siamese-BERT-wwm-CNN-Bi-LSTM	80.80	81.03	80.91
3	Siamese-BERT-wwm-CNN + Bi-LSTM	86.41	86.12	86.27
4	Siamese-BERT-wwm-CNN	86.65	86.18	86.41
5	Siamese-BERT-wwm-Bi-LSTM	<b>87.30</b>	<b>88.35</b>	<b>87.82</b>

Bold values refer to the maximum prediction performance in each circumstance.

to CNN-Bi-LSTM in model 2. CNN-Bi-LSTM refers to using Bi-LSTM to get the long-range features from the extracted local features from CNN [41]. CNN + Bi-LSTM splices the long-range features from Bi-LSTM and local features from CNN as a new representation of features [42, 43].

As shown in Table 9, replacing Bi-LSTM or CNN with the combination of Bi-LSTM and CNN cannot improve mapping relationship discovery performance. On the one hand, it may be that a little semantic information extracted by BERT-wwm remains after the multilayer neural network. On the other hand, it may be that the elements are too short to contain adequate information. There is an overlap between long-range features and local features for the short text, resulting in large weights for some features and weakening the model performance.

## 5. Conclusions

Mapping relationship discovery for multi-dimensional architectures in ATS is essential for theoretical research. It makes sense to accurately and quickly identify the pairs of elements that have mapping relationships. This study develops a method based on Siamese-BERT-wwm-Bi-LSTM for mapping relationship discovery to alleviate the high subjectivity and underutilization of textual data. The method automates the mapping relationship discovery of multidimensional architectures. According to the experimental results, the text-matching models are feasible for the mapping relationship discovery task. By comparing the results of distinct models, Siamese-BERT-wwm-Bi-LSTM achieves superior performance in the metrics of precision rate, recall rate, and  $F_1$ -score.

In this paper, we apply various models for mapping relationship discovery. By analyzing the results of the models, it is concluded that Siamese-BERT-wwm-Bi-LSTM has a better performance. And the ablation experiments verify the validity of each module of the hybrid model. Also, we explore the applicability of the joint use of Bi-LSTM and CNN in the scenario of this paper. The combination of Bi-LSTM and CNN leads to performance degradation, and it is better to use either model alone.

The preprocessing method in this paper for textual data of the multidimensional architectures can improve. When performing text feature enhancement, a random negative sampling method is used to expand the data. However, this approach does not fully utilize the information from negative samples in practice. In future studies, we will improve

the way of negative sampling. At the same time, we will try to enhance the text features through text generation.

## Data Availability

The textual data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This research was funded in part by the National Key R&D Program of China (grant no. 2020YFB1600400) and Innovation-Driven Project of Central South University (grant no. 2020CX041).

## References

- [1] Y. Li, S. E. Li, X. Jia, S. Zeng, and Y. Wang, "FPGA accelerated model predictive control for autonomous driving," *Journal of Intelligent and Connected Vehicles*, vol. 5, no. 2, pp. 63–71, 2022.
- [2] L. Yue, M. Abdel-Aty, and Z. Wang, "Effects of connected and autonomous vehicle merging behavior on mainline human-driven vehicle," *Journal of Intelligent and Connected Vehicles*, vol. 5, no. 1, pp. 36–45, 2022.
- [3] T. Olovsson, T. Svensson, and J. Wu, "Future connected vehicles: communications demands, privacy and cybersecurity," *Communications in Transportation Research*, vol. 2, no. 100, Article ID 100056, 2022.
- [4] N. Coretti Sanchez, I. Martinez, L. Alonso Pastor, and K. Larson, "On the simulation of shared autonomous micro-mobility," *Communications in Transportation Research*, vol. 2, no. 100, Article ID 100065, 2022.
- [5] L. L. You, F. Zhao, L. Cheah, K. Jeong, P. C. Zegras, and M. Ben-Akiva, "A generic future mobility sensing system for travel data collection, management, fusion, and visualization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4149–4160, 2020.
- [6] L. L. You, B. Tuncer, and H. Xing, "Harnessing multi-source data about public sentiments and activities for informed design," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 343–356, 2019.
- [7] L. L. You, J. S. He, W. Wang, and M. Cai, *Autonomous Transportation Systems and Services Enabled by the Next-Generation Network*, IEEE Network, Manhattan, NY, USA, 2022.
- [8] K. Zhang, H. Liu, and D. M. Liu, *Construction Method and Application of Intelligent Transportation System Framework*, China Communications Press, Beijing, China, 2013.

- [9] X. J. Wang, Z. W. Zhang, and T. Y. Qi, *China Intelligent Transportation System Framework*, China Communication Press, Beijing, China, 2003.
- [10] X. J. Wang, "Study on the framework of intelligent transportation system," *Computer and Communications*, vol. 19, no. 4, pp. 4–7, 2001.
- [11] K. Zhang, T. Y. Qi, D. M. Liu, C. Y. Wang, R. H. He, and H. Liu, "The latest achievements of Chinese national ITS architecture," *Journal of Transportation Systems Engineering and Information Technology*, vol. 5, no. 5, pp. 6–11, 2005.
- [12] Y. S. Su and S. Q. Ni, "Research on the structure of intelligent railway system in China," *Journal of Transportation Systems Engineering and Information Technology*, vol. 15, no. 4, pp. 135–140, 2017.
- [13] J. Niu, "Research standards for iot architectures for intelligent transportation systems," *China Standardization*, vol. 14, no. 1, pp. 231–232, 2017.
- [14] W. Wei, L. Zheng, and M. Cai, "Research on user needs of Intelligent Transportation System for autonomous transportation," *Technology and Economy in Areas of Communications*, vol. 24, pp. 1–6, 2022.
- [15] Z. Wang, L. M. Jia, Y. H. Wang, Y. Qin, and X. Q. Cheng, "Structural design method of railway intelligent transportation system," *China Railway Science*, vol. 27, no. 2, pp. 116–119, 2006.
- [16] J. J. Tang, Y. Liu, Q. Fu, and H. N. Tuo, "Multi-dimensional mapping relationship modeling in autonomous transportation system Based on Fuzzy Theory," *Forest engineering*, vol. 38, no. 2, pp. 146–153, 2022.
- [17] X. H. He, *Study on Key Issues in the Framework Design of Inland River Shipping System*, Tongji University, Shanghai, China, 2007.
- [18] J. Ma, "Research based on multi-level mapping relationship model for dynamic system based on medium granularity," *Journal of Systems Science and Mathematical Sciences*, vol. 38, no. 8, pp. 886–880, 2018.
- [19] C. L. Wang, Y. ., H. Yang, and F. Deng, "A review of text similarity approaches," *Information Science*, vol. 37, no. 3, pp. 158–168, 2019.
- [20] R. Maipradit, H. Hata, and K. Matsumoto, "Sentiment classification using N-gram inverse document frequency and automated machine learning," *IEEE Software*, vol. 36, no. 5, pp. 65–70, 2019.
- [21] M. Ayub, M. A. Ghazanfar, and M. Maqsood, "A Jaccard base similarity measure to improve performance of CF based recommender systems," in *Proceedings of the 32nd International Conference on Information Networking(ICOIN)*, pp. 1–6, Wuhan, China, April 2018.
- [22] Y. Chen, *Convolutional Neural Network for Sentence Classification*, University of Waterloo, Waterloo, UK, 2015.
- [23] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "Abcnn: attention-based convolutional neural network for modeling sentence pairs," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 259–272, 2016.
- [24] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," 2016, <https://arxiv.org/abs/1605.05101>.
- [25] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5–6, pp. 602–610, 2005.
- [26] H. Palangi, L. Deng, and Y. Shen, "Semantic modelling with long-short-term memory for information retrieval," 2014, <https://arxiv.org/abs/1412.6629>.
- [27] Q. Chen, X. Zhu, and Z. Ling, "Enhanced LSTM for natural language inference," 2016, <https://arxiv.org/abs/1609.06038>.
- [28] T. Mikolov, K. Chen, and G. Corrado, "Efficient estimation of word representations in vector space," 2013, <https://arxiv.org/abs/1301.3781>.
- [29] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing(EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014.
- [30] J. Devlin, M. W. Chang, and K. Lee, "Bert: pre-training of deep bidirectional transformers for language understanding," 2018, <https://arxiv.org/abs/1810.04805>.
- [31] Y. Liu, M. Ott, and N. Goyal, "Roberta: a robustly optimized bert pretraining approach," 2019, <https://arxiv.org/abs/1907.11692>.
- [32] Z. Lan, M. Chen, and S. Goodman, "Albert: a lite bert for self-supervised learning of language representations," 2019, <https://arxiv.org/abs/1909.11942>.
- [33] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for Chinese bert," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3504–3514, 2021.
- [34] N. Reimers and I. Gurevych, "Sentence-bert: sentence embeddings using siamese bert-networks," 2019, <https://arxiv.org/abs/1908.10084>.
- [35] J. Dong, F. He, and Y. Guo, "A commodity review sentiment analysis based on BERT-CNN model," in *Proceedings of the 2020 5th International Conference on Computer and Communication Systems(ICCCS)*, pp. 143–147, Shanghai, China, May 2020.
- [36] S. Agrawal, S. Dutta, and B. K. Patra, "Sentiment analysis of short informal text by tuning BERT-Bi-LSTM model," in *Proceedings of the IEEE EUROCON 2021-19th International Conference on Smart Technologies, SmartTech-IC*, pp. 98–102, Lviv, Ukraine, July 2021.
- [37] H. Choi, J. Kim, and S. Joe, "Evaluation of bert and albert sentence embedding performance on downstream nlp tasks," in *Proceedings of the 2020 25th International Conference on Pattern Recognition(ICPR)*, pp. 5482–5487, Mico, Italy, May 2021.
- [38] D. Viji and S. Revathy, "A hybrid approach of Weighted Fine-Tuned BERT extraction with deep Siamese Bi-LSTM model for semantic text similarity identification," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 6131–6157, 2022.
- [39] C. Liu, C. Zhu, and W. Zhu, "Chinese named entity recognition based on BERT with whole word masking," in *Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence*, pp. 311–316, Tianjin, China, April 2020.

- [40] J. Zhang, Y. Li, and J. Tian, "LSTM-CNN hybrid model for text classification," in *Proceedings of the 3rd Advanced Information Technology, Electronic and Automation Control Conference(IAEAC)*, pp. 1675–1680, Chongqing, China, October 2018.
- [41] Y. Luan and S. Lin, "Research on text classification based on CNN and LSTM," in *Proceedings of the IEEE international conference on artificial intelligence and computer applications(ICAICA)*, pp. 352–355, Dalian, China, March 2019.
- [42] N. Jin, C. J. Zhao, H. R. Wu, Y. S. Miu, H. C. Yang, and B. Z. Yang, "Agricultural short text matching technology Based on multi-semantic features," *Transactions of the Chinese Society for Agricultural*, vol. 53, no. 5, pp. 325–331, 2022.
- [43] W. Li, S. Gao, and H. Zhou, "The automatic text classification method based on bert and feature union," in *Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems(ICPADS)*, pp. 774–777, Beijing, China, December 2019.